Enershare
The Energy Data Space for Europe

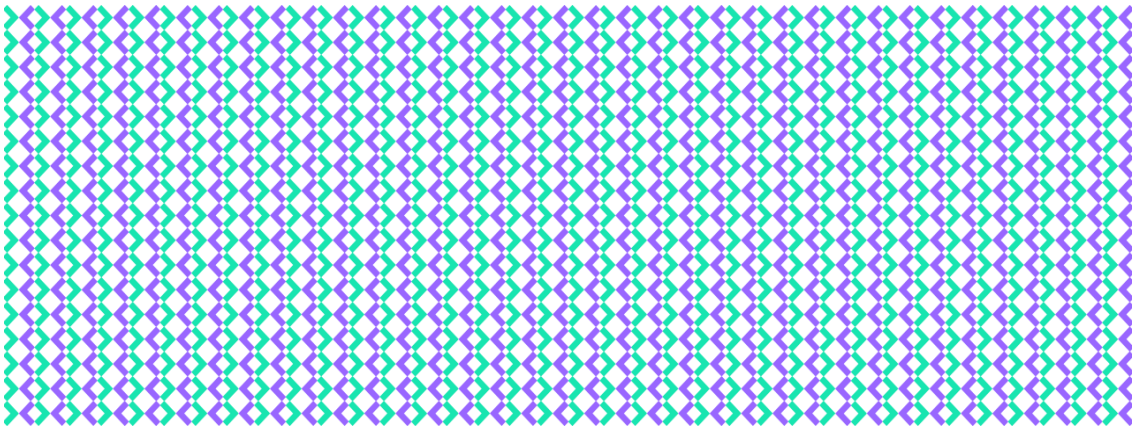
European Common Energy Data Space Framework Enabling Data Sharing - Driven Across – and Beyond – Energy Services

enershare.eu

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

D4.3 ENERSHARE Trust and sovereignty building blocks

Final version



Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Publication details

Grant Agreement Number	101069831
Acronym	ENERSHARE

Full Title	European Common Energy Data Space Framework Enabling Data Sharing-Driven Across — and Beyond — Energy Services
Topic	HORIZON-CL5-2021-D3-01-01 'Establish the grounds for a common European energy data space'
Funding scheme	HORIZON-IA: Innovation Action
Start Date	Jul 1, 2022
Duration	36 months
Project URL	enershare.eu
Project Coordinator	Engineering
Deliverable	D4.3 – ENERSHARE Trust and sovereignty building blocks (Final version)
Work Package	WP4 – Trust and sovereignty enabling framework and building blocks
Delivery Month (DoA)	M26
Version	1.0
Actual Delivery Date	August 30, 2024
Nature	Report
Dissemination Level	PU

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Lead Beneficiary	TNO
Authors	Sonia Jimenez (IDSA), Lorenzo Cristofori and Alessandro Rossi (ENG), Rizwan Mehmood (FhG), Maarten Kollenstart (TNO), Michiel Stornebrink (TNO), Arjan Stoter (TNO)
Quality Reviewer(s)	Apostolos Kapetanios (ED), Marzia Mammina, and Alessandro Rossi (ENG)
Keywords	Data space connector, identity and access management, usage control, trust, sovereignty, full stack integrity, blockchain, ledger

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Document History

Ver.	Date	Description	Author	Partner
0.1	April 11, 2024	Table of content	Michiel Stornebrink	TNO
0.2	May-Jun	Writing phase	All	ENG FhG IDSA INESC TNO
0.3	July 10, 2024	Ready for review	Arjan Stoter Michiel Stornebrink	TNO
0.4	July 31, 2024	Reviewed	Marzia Mammina Alessandro Rossi Apostolos Kapetanios	ENG ED
0.5	Aug, 15, 2024	Review processed	Arjan Stoter and WP participants	TNO
1.0	Aug 30, 2024	Final version	Arjan Stoter Michiel Stornebrink	TNO

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Table of Contents

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Table of Figures

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

1 Introduction

1.1 About the project

The overall vision of ENERSHARE is to develop and demonstrate a European Common Energy Data Space which will deploy an intra-energy and cross-sector interoperable and trusted energy data ecosystem. Private consumers, energy- and non-energy business stakeholders and regulated operators will be able to access, share and reuse, based upon voluntary agreements or legal obligations where such obligations are in force: (a) large sources of currently fragmented and dispersed data; (b) data-driven cross-value chain services and digital twins for various purposes.

1.2 Work package objectives and tasks

Previous ENERSHARE deliverable D4.1 – "ENERSHARE trust and sovereignty building blocks (Alpha version)" addressed the state of the art and the pilot requirements in the energy dataspace in terms of identity- and access management and usage control policies and, additionally, discussed full-stack integrity and distributed ledger technology. ENERSHARE deliverable D4.2 – "ENERSHARE trust and sovereignty building blocks (Beta version)" described the (extended) designs for trust and sovereignty related components for the ENERSHARE dataspace and describes the implementation and validation plans towards third technology release.

Two essential components were provided by WP4 for the first technology release of the energy dataspace (i.e. MVP-1), namely the identity provider and the dataspace connector implementation.

It was concluded in deliverable D4.1 that WP4 would continue the work on trust and sovereignty by:

(1) integrating self-sovereign identity (SSI) concepts into the identity manager component (T4.2),
(2) implementing usage control policies that were identified in the pilots (T4.3),
(3) integrating full stack integrity in the TNO Security Gateway (TSG) connector implementation (T4.1), and
(4) implementing a component using distributed ledger technology (T4.4).

Figure 1 shows how these components and tasks relate.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

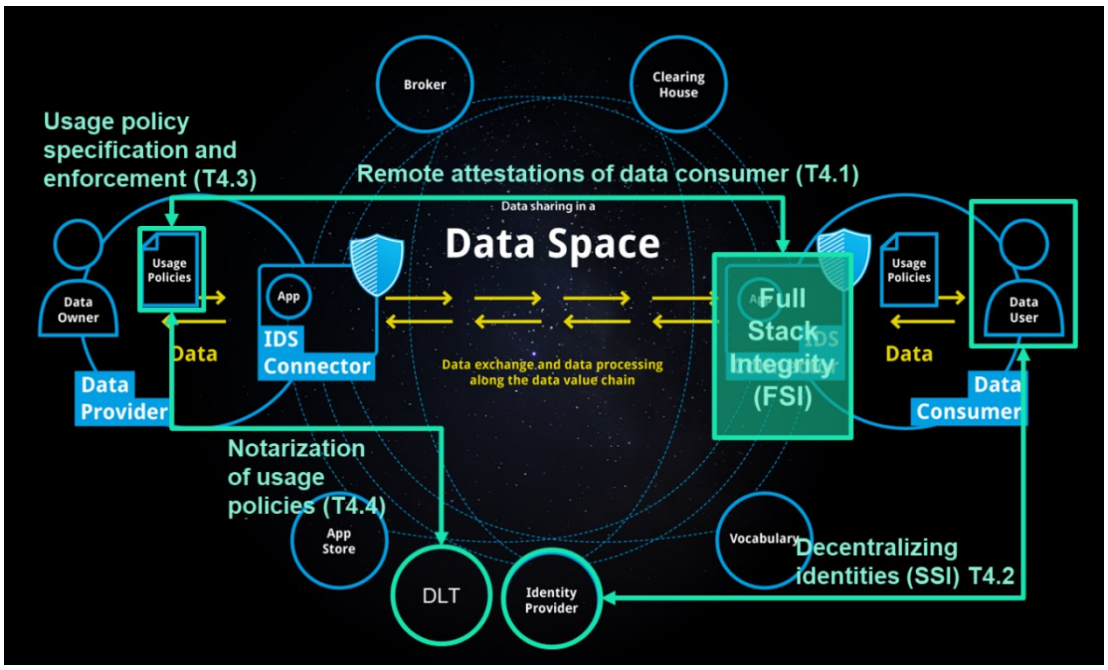


Figure 1: Extended trust and sovereignty functionalities in the ENERSHARE Data Space.

1.3 About this document

ENERSHARE deliverable D4.3 – "Trust and sovereignty building blocks, final version" is the third deliverable of a series of three ENERSHARE deliverables that relate to trust and sovereignty in components for the ENERSHARE Data Space. Following the previous alpha version (D4.1) and beta version (D4.2), D4.3 describes the final result of the joint activities in ENERSHARE WP4, resulting in (extended) software components that were provided to WP8 for integration in the energy dataspace in the final technology release.

The current deliverable describes the final design and implementation of decentralized identity management, usage control enforcement, notarization on the blockchain, and remote attestation for full-stack integrity in the ENERSHARE dataspace.

1.4 Intended audience

The current deliverable is intended for the following audience:

- All project partners that are involved in WP4. This document contains the lower-level designs of the building blocks for the third technology release
- All project partners from other WPs (especially WP5). This document contains the scope and building blocks that are considered part of the trust and sovereignty framework. It

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

allows others to understand what building blocks and functionalities can and cannot be expected from WP4.

- All project partners involved in integration (WP8) and pilot implementations (WP9). This document addresses what building blocks are provided as part of the final technology release and how and which pilot requirements are addressed.
- External stakeholders of the project that would like to be part of the ENERSHARE energy dataspace and need to integrate and deploy trust and sovereignty building blocks.

1.5 Reading recommendations

The remainder of the document is structured as follows:

- **Chapter 2 Architecture** describes the overall position of the final version of the trust and sovereignty building blocks inside the ENERSHARE reference architecture, as well as their mapping to the Data Spaces Support Centre (DSSC) building blocks, and, additionally provides a short description of the Dataspace Protocol and its relation to the trust and sovereignty building blocks.
- **Chapter 3 Decentralized identity management** explores the shift towards decentralized identities in the ENERSHARE dataspace, including the functional/logical view, technical design, and next steps to take towards the following technology release. Also, the alignment with sister projects and the importance of interoperability are discussed.
- **Chapter 4 Usage control enforcement** details the functional/logical view of usage control policies, including policy specification, classes, and components. It also describes the technical design and current developments in usage control enforcement, focusing on technology selection and pilot policy requirements
- **Chapter 5 Usage policy notarization on the blockchain** outlines how Distributed Ledger Technology (DLT) and blockchain can be leveraged to notarize usage policies, ensuring their immutability and non-repudiation, which are critical for establishing a secure and trusted energy dataspace. The chapter details the use of smart contracts, specifically designed for the Proof of Existence (PoE) of documents or policies. Finally, the current implementation of the PoE is laid out, though final implementation is planned for D4.3.
- **Chapter 6 Remote attestation for full stack integrity** discusses the process that allows one party (the verifier) to verify the integrity of the software and hardware environment of another party (the prover). The chapter goes on to discuss the technical design of the remote attestation mechanism, including the use of sidecar architecture to facilitate the attestation process.
- **Chapter 7 Conclusions** summarizes the updated set of software components for this final release.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

2 Architecture

The current deliverable addresses the final design and implementation of the following ENERSHARE trust and sovereignty building blocks:

- Decentralized identity management,
- Usage control enforcement,
- Usage policy notarization on the blockchain, and
- Remote attestation for full-stack integrity in the ENERSHARE dataspace.

This chapter describes the overall position of the final version of the above mentioned building blocks inside the ENERSHARE reference architecture, as well as their mapping to the DSSC building blocks.

In addition, this chapter gives a short description of the Dataspace Protocol and its relation to the trust and sovereignty building blocks.

2.1 Positioning in project reference architecture

ENERSHARE deliverable D2.5 – "Final Version of the ENERSHARE Requirements, SSH-driven Approach and Reference Architecture" described the Reference Architecture for the ENERSHARE project, as depicted in Figure 2.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

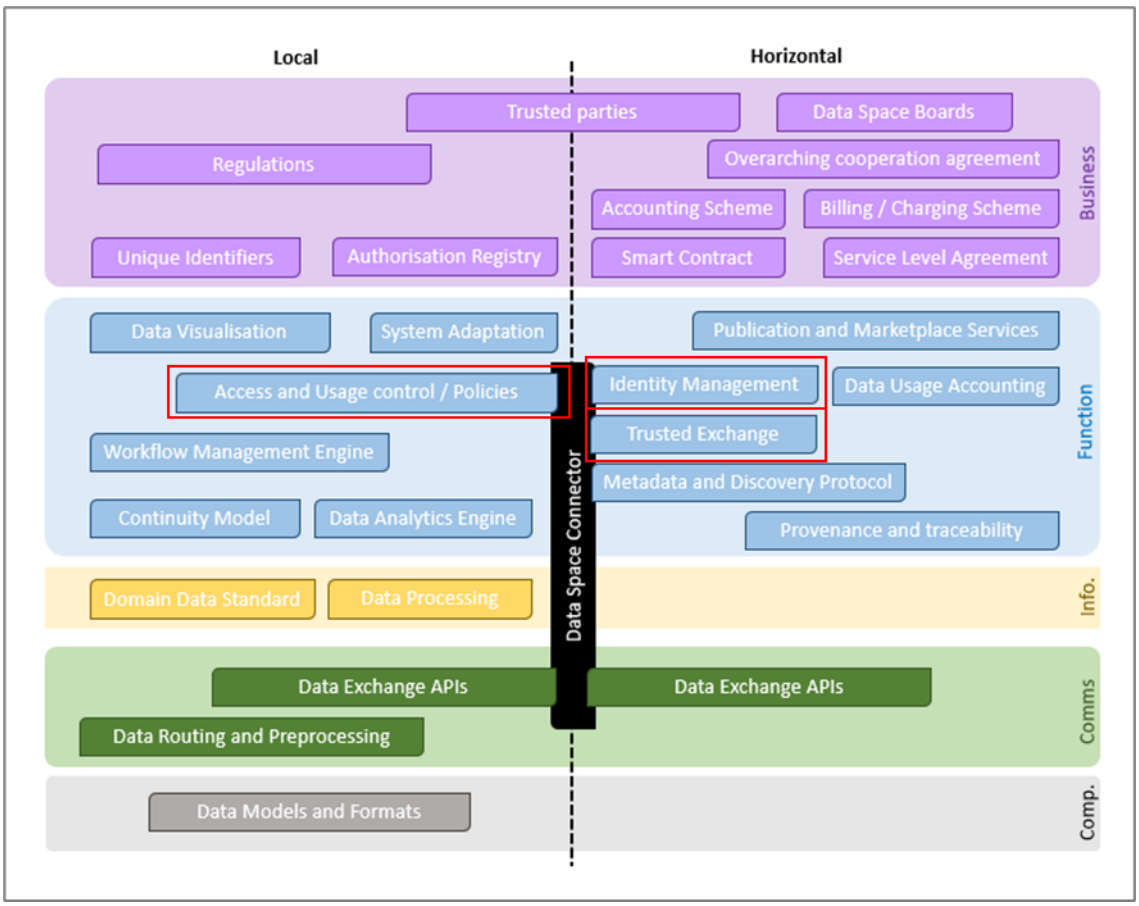


Figure 2: ENERSHARE dataspace reference architecture.

The trust and sovereignty building blocks that are addressed in the current deliverable (decentralized identity management, usage control enforcement, usage policy notarization on the blockchain, and remote attestation for full-stack integrity) relate to the functional level of the reference architecture -highlighted in red in Figure 2- and address “Identity Management”, “Access- and Usage control / Policies”, and “Trusted Exchange”.

The Reference Architecture in ENERSHARE deliverable D2.5 adopted the 4+1 Architectural View Model (Figure 3), containing the Logical View, Development View, Process View, and Deployment- or Physical View, and incorporated the use cases described in ENERSHARE deliverable D2.1 – “Use cases’ descriptions and list of minimum Data Space building blocks required for pilots” as input for the Scenarios. The 4+1 Architectural View Model (Figure 3) was also used to describe each of the trust and sovereignty building blocks in the current deliverable in chapters 3, 4, 5, and 6.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

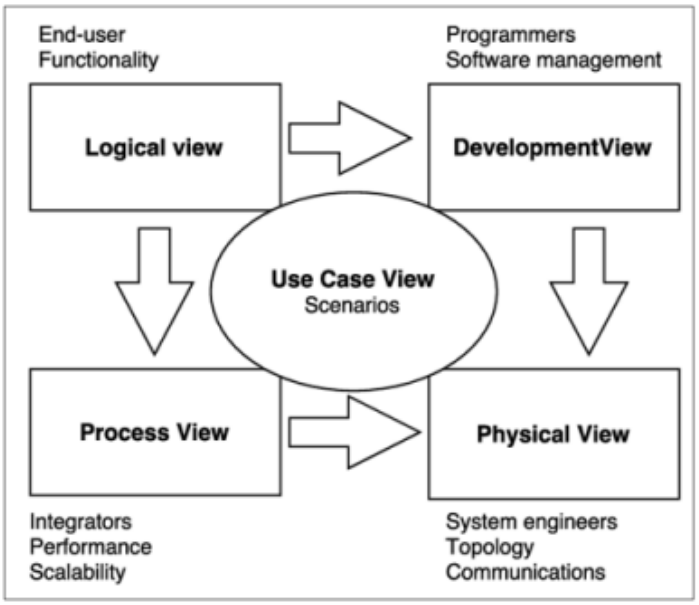


Figure 3: Extended trust and sovereignty functionalities in the ENERSHARE dataspace

2.2 Positioning in DSSC Blueprint

The OPEN DEI building blocks[1] have served as a reference for dataspaces over the past years. At the time of writing, and as described in ENERSHARE deliverable D4.2, these building blocks have been adopted by the DSSC, resulting in the DSSC Blueprint version 1.0[2] with its own set of building blocks (Figure 4).

The ENERSHARE project has adopted the DSSC building blocks. Compared to the OPEN DEI building blocks, the DSSC building blocks include (1.) the involvement of a wider range of stakeholders, such as Small and Medium Sized Enterprises (SMEs), startups, and public administrations in the DSSC's Blueprint to create a more robust and diverse data ecosystem; (2.) a strong emphasis on interoperability between different dataspaces, leveraging insights from various Common Service Areas (CSAs) to facilitate smoother data exchange and integration across sectors, enhancing the overall utility of the dataspace; (3.) an inherently higher focus on innovation and scalability, drawing from a variety of CSAs, each bringing unique insights and advancements; and finally, (4.) a more comprehensive governance model that addresses issues like data sovereignty, privacy, and security in more depth, the latter being reflected in the trust and sovereignty building blocks described in the current document.

[1] https://design-principles-for-data-spaces.org/

[2] https://dssc.eu/space/News/blog/381878275/Introducing+Blueprint+1.0%3A+the+evolution+of+Data+Spaces

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

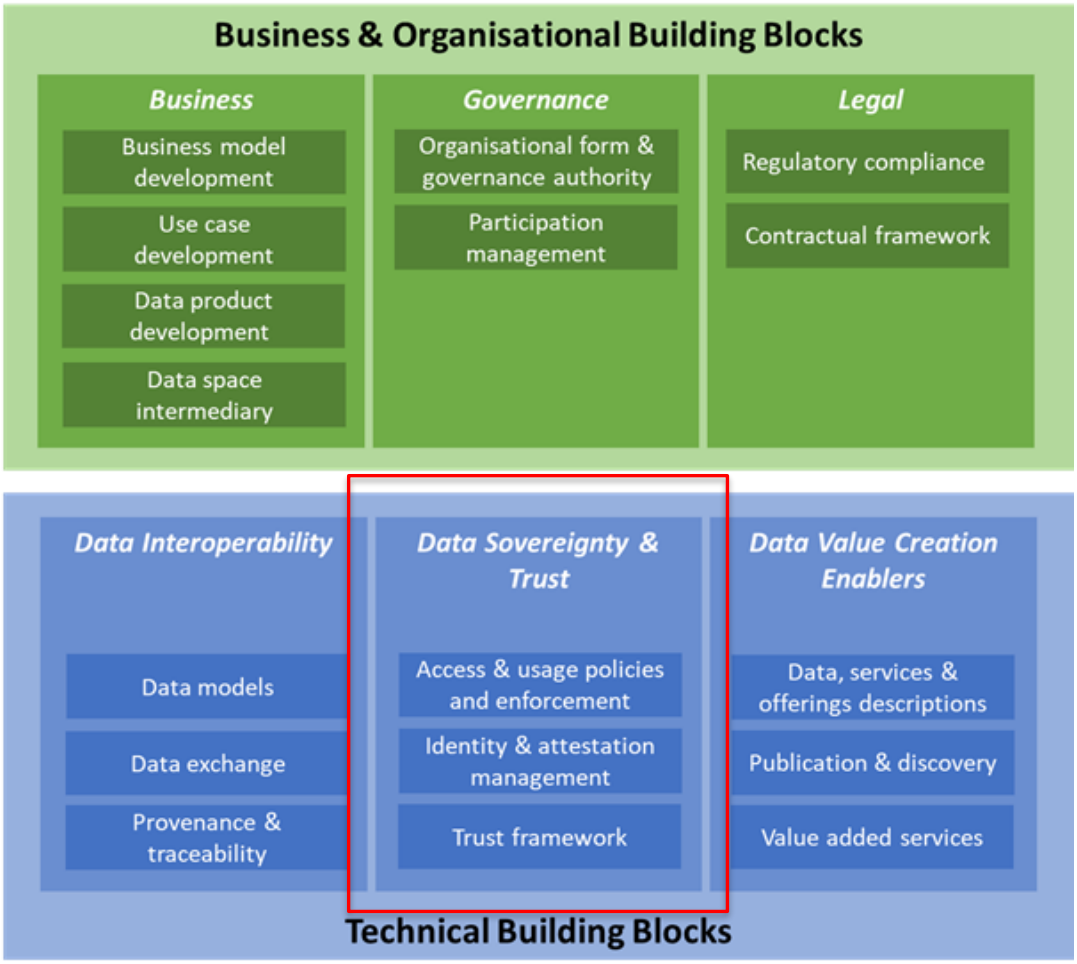


Figure 4: ENERSHARE trust and sovereignty positioned in the DSSC building blocks

ENERSHARE deliverable D8.2 – "ENERSHARE Data Space (1st Technology Release)" described a mapping of the ENERSHARE components to both the OPEN DEI building blocks and DSSC building blocks. As can be seen in Figure 4 (marked in red), the ENERSHARE trust and sovereignty building blocks map to the DSSC technical building blocks "Access & usage policies and enforcement", "Identity & attestation management", and "Trust framework".

2.3 Dataspace Protocol

ENERSHARE deliverable D4.2 provided a description of the Dataspace Protocol, being a set of specifications, schemas, and protocols for publishing data, usage negotiation, and data access in a dataspace, which will supersede the existing International Data Spaces (IDS) protocol.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Since deliverable D4.2, the first full version of the Dataspace Protocol was released in February 2024 as version 2024-1[3]. However, the technical description of this full version, and its impact on the ENERSHARE dataspace is similar to the preliminary version of the Dataspace Protocol that was described in D4.2.

What has changed since ENERSHARE deliverable D4.2 is that the Eclipse Dataspace Protocol[4] has been established, under the umbrella of the Eclipse Dataspace Working Group[5]. This organizational change in governance structure could allow the Dataspace Protocol to evolve independently and eases the path towards international standardization of the protocol.

The adoption of the Dataspace Protocol (and thereby the transition from the previously used IDS protocol) is executed in a separate environment within the ENERSHARE project, parallel to the existing IDS Communication Guide based environment. This is to allow ENERSHARE partners to migrate to the Dataspace Protocol according to their own timeline. This is especially relevant for ENERSHARE partners that create services that are tightly integrated with the dataspace, e.g., the marketplace, the app store, and for which the transition to the Dataspace protocol has the highest impact.

2.4 Work Package deliverable iterations

The current deliverable D4.3 “ENERSHARE trust and sovereignty building blocks (Final version)” is the third of three deliverables, preceded by D4.1 and D4.2. Deliverable D4.1 “ENERSHARE trust and sovereignty building blocks (Alpha version)” described the requirements for the ENERSHARE trust and sovereignty building blocks, followed by D4.2 (beta version), containing these building blocks’ extended design. D4.3 accompanies the final implementation of the trust and sovereignty building blocks in ENERSHARE, describing the final architecture and code repositories of each of the trust and sovereignty building blocks.

[3] https://github.com/International-Data-Spaces-Association/ids-specification/releases/tag/2024-1

[4] https://projects.eclipse.org/projects/technology.dataspace-protocol-base

[5] https://dataspace.eclipse.org/

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

3 Decentralized identity management

3.1 Introduction

Throughout the project, three Identity and Access Management (IAM) solutions have been implemented. At the beginning of the project, various IAM building blocks were identified and evaluated for their suitability. For the initial Minimal Viable Product (MVP), ENERSHARE implemented a centralized approach using IAM components from the IDS Reference Architecture Model developed by TNO, specifically the TSG Certificate Authority (CA) and the Dynamic Attribute Provisioning Service (DAPS). The details of the initial implementation were described in D4.1.

The Market place developed in WP5 and the App Store from WP6 require the identification of human users, therefore a complementary OAuth 2.0 and OpenID compliant Identity Provider was implemented, as detailed in chapter 3.7.

As part of the work in T4.2 and to ensure alignment with sister projects, a separate environment was established where IAM based on decentralized identities was implemented, as described in this chapter.

3.2 Scenario view

The scenarios related to decentralized identity management fall largely in two phases:

1. the issuance phase
2. the usage phase.

In the issuance phase (1), a holder wants to get a credential issued by a trusted issuer. In many situations, the issuer of the credentials will be a data space authority to indicate that the holder is part of the dataspace and agreed upon the set of common rules agreed upon in the dataspace. The issuer can, however, also be any other party that can make claims about the holder that might be relevant for usage within the dataspace.

In the usage phase (2), a verifier requires the holder to present the relevant credentials for the specific interaction between the holder and the verifier. For example, the verifier might need to verify that the holder has agreed upon the common set of rules before data is exchanged between the two parties. In Figure 5, the overview of the scenarios and related actors is shown.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

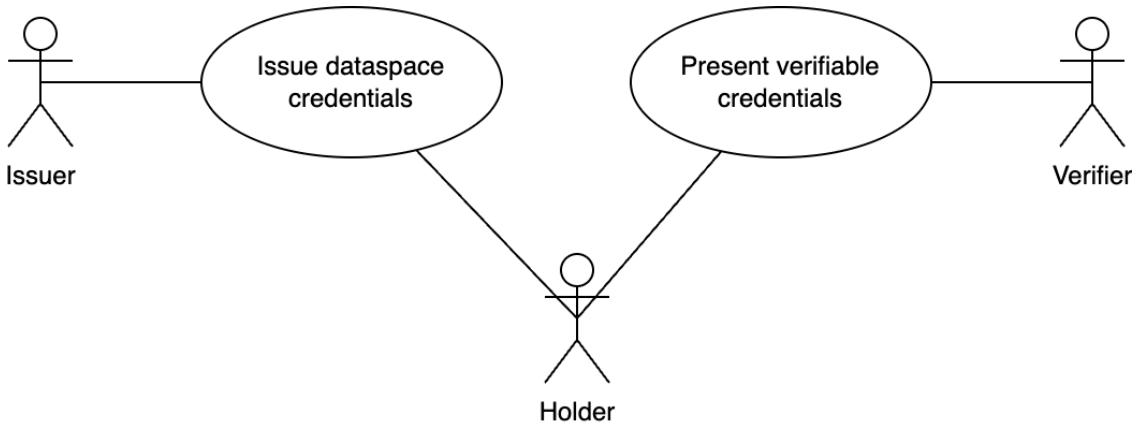


Figure 5: Decentralized identity management scenario overview

3.2.1 Issue dataspace credentials

This scenario involves creating a verifiable credential for a participant (the holder) from a dataspace authority (the issuer), or an entity authorized to issue verifiable credentials for the dataspace.

Steps:

1. **Establish relationship between the holder and issuer**. The issuer of credentials needs to verify the holder based on the governance rules from the dataspace. The issuer must at least receive the Decentralized Identifier (DID) identifier from the holder to issue the credential.
2. **Create credential offer**. The issuer creates a template for the credential that will be handed out to the holder, containing all relevant information that should be part of the verifiable credential. A pre-authorization code will be shared with the holder, which can be used to retrieve the credential. The way the pre-authorization code is shared can be done via any communication method, e.g. via mail.
3. **Credential request preparation**. The holder prepares the credential request by retrieving the metadata of the issuer and by requesting an access token based on the pre-authorization code received.
4. **Request credential**. The holder requests the credential from the issuer.
5. **Credential validation**. The holder verifies the received credential to ensure it is valid and issued by the right issuer.

Outcome: A verifiable credential has been issued to the holder indicating the holder is participating in the dataspace.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

3.2.2 Present verifiable credentials

This scenario outlines the presentation of verifiable credentials by two dataspace participants, the holder and the verifier. The verifier will be able to request specific credentials based on its needs for establishing trust. The scenario focuses on a single presentation interaction between the holder and verifier, but the scenario can be executed both ways to ensure mutual trust between the two parties.

Steps:

1. **Request holder token**. The holder will request an ID token to be created that allows the receiver of the token to initiate the presentation request flow.
2. **Presentation request preparation**. The verifier will validate the holder's ID token and requests a similar ID token from the verifier side, with a link to the holder's ID token. In addition, the presentation service endpoint will be fetched from the DID document of the holder. Also, the required presentation definition, including the information what kind of information the verifier needs, will be created.
3. **Presentation request**. The verifier will request a presentation based on verifier's ID token and the presentation definition at the presentation service endpoint of the holder.
4. **Verifiable Presentation creation**. The holder will validate the verifier's ID token and fetches the required verifiable credentials needed to fulfill the presentation definition from the verifier.
5. **Verifiable Presentation validation**. The verifier will validate the verifiable presentation, ensuring the presentation originated from the right holder and is cryptographically signed with the right key material. Also, the verifiable credentials within the presentation are evaluated to ensure they match the presentation definition.

Outcome: The verifier has received a correct verifiable presentation that fulfills the requirements for establishing trust between the verifier and the holder. This forms the basis of any interactions between the holder and verifier.

3.3 Logical view

The logical view of decentralized identity management can be largely covered by the swim lane diagram in Figure 6. Given the decentralized nature, the interactions between the actors provide the most meaningful overview of the functionality related to the issuance and usage of verifiable credentials and presentations.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

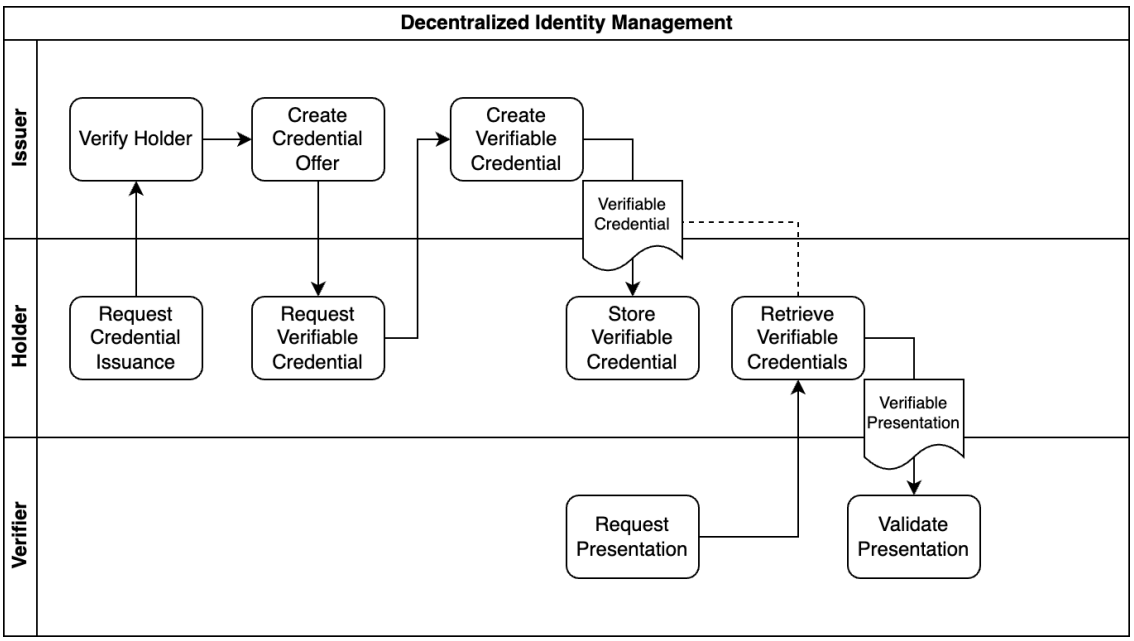


Figure 6: Logical overview of issuance and usage phase for Decentralized Identity Management

The initial interaction between the holder and issuer can be executed in different ways depending on the rules and governance of the dataspace. In this final version of the building blocks for identity management, the onboarding of new participants is an off-line process. In the software, it is assumed there is a prior relationship between the holder and issuer, e.g. by a contract that a participant signs with the dataspace authority. During the verification of the holder, the issuer is required to validate all relevant information that is available of the holder.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

3.4 Process view

3.4.1 Issue dataspace credentials

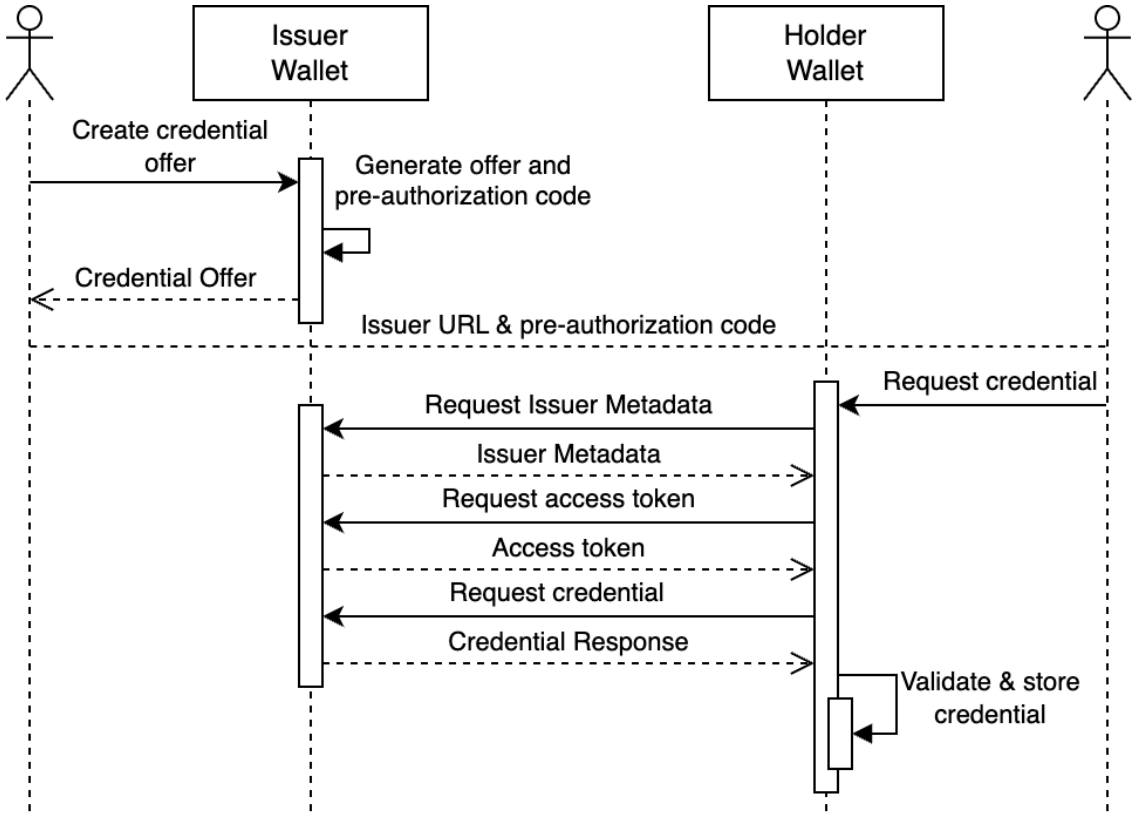


Figure 7: Issue dataspace credentials sequence diagram

1. Create a credential offer with the relevant information of the credential subject properties that should be created by the wallet.
2. Create an internal credential offer representation combined with a pre-authorization code that can be shared with the holder.
3. Return the credential offer representation to the user.
4. Exchange the Issuer URL, i.e. the root URL of the wallet that from which the `/.well-known/openid-credential-issuer` metadata endpoint can be retrieved, and the pre-authorization code with the holder. This exchange does not require any software system to be in place, allowing this exchange to be able to execute in-person but also via additional systems like email.
5. Based on the Issuer URL and pre-authorization code the user requests the credential issuance flow to be started from its wallet.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

6. Request Issuer Metadata according to https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html#name-credential-issuer-metadata.
7. Response of Issuer Metadata.
8. Request an access token based on the pre-authorization code.
9. Response of an access token with a specific scope of requesting the credential that belongs to the credential offer.
10. Request the credential with the access token and optionally proof of possession of key material that the credential should be bound to.
11. Validate the access token and proof of possession and create the actual verifiable credential. The response will be packaged in a verifiable presentation signed as JSON Web Token.

3.4.2 Present verifiable credentials

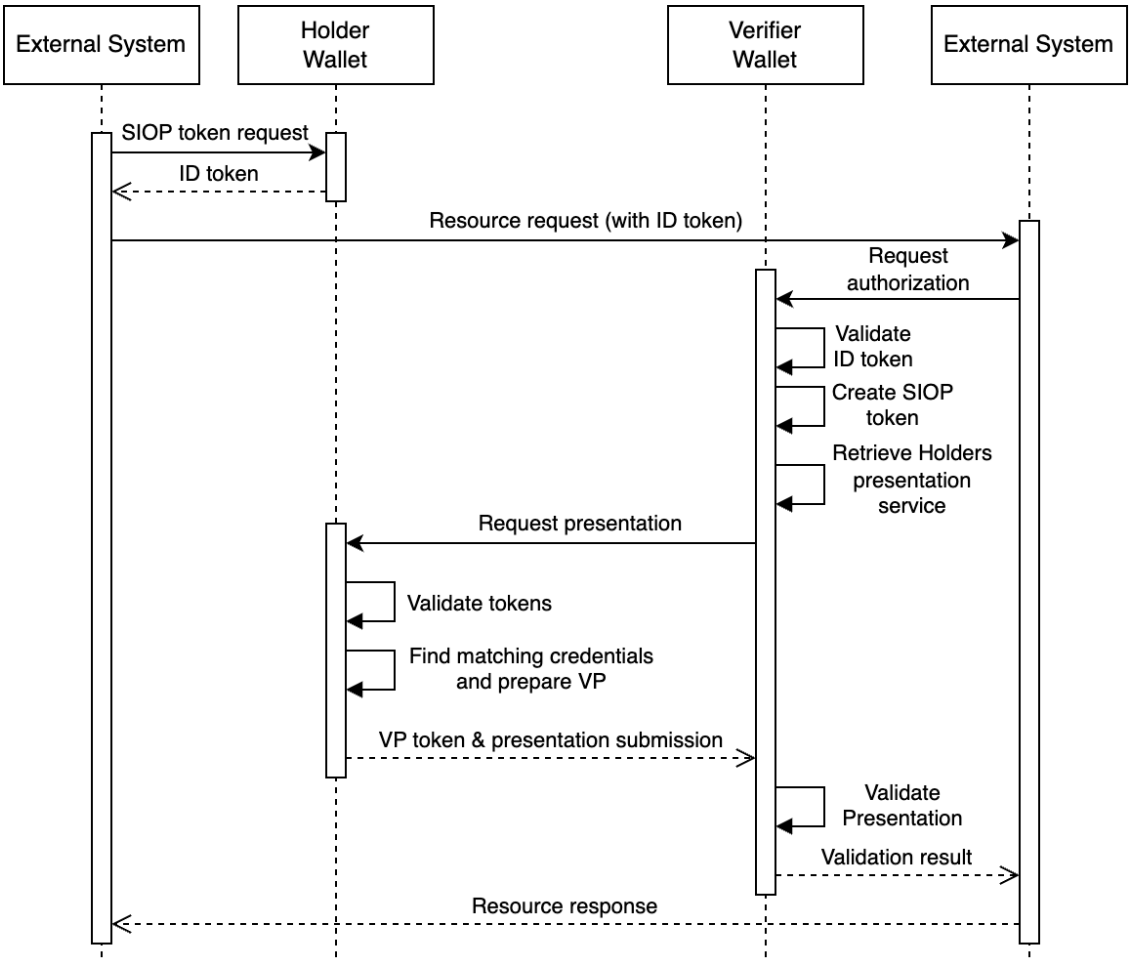


Figure 8: Present verifiable credentials sequence diagram

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

1. Request a Self-Issued ID token targeted at the verifier (*audience*) with an automatically generated access token for the verifier to request the presentation, with an optional scope to limit the access to certain credentials.
2. ID token signed with the default key defined in the wallet.
3. Resource request (e.g. Data Space Protocol Request) with the ID token.
4. Verification request with the holder's ID token and a presentation definition according the DIF Presentation Definition specification[6].
5. Validation of the holders ID token, which, additionally, requires resolving the DID document of the holder to retrieve the public key material used to sign the ID token.
6. Create a Self-Issued ID token targeted at the holder (*audience*) incorporating the access token from the holder's ID token.
7. Retrieve the `"Presentation"` service from the holder's DID document to find the service endpoint for requesting the presentation.
8. Request the presentation at the service endpoint with the presentation definition and the ID token.
9. Validate the verifier's ID token and the access token inside the ID token.
10. Find matching credentials based on the presentation definition and the scope of the access token. And generate a presentation submission according the DIF Presentation Submission specification[7].
11. Return the Verifiable Presentation in JSON Web Token (JWT) format accompanied by the presentation submission.
12. Validate the Verifiable Presentation and validate whether it matches the requested presentation definition.
13. Return the Verifiable Presentation when all checks are successful.
14. Response to the original resource request.

3.5 Development view

The development view of decentralized identity management focuses on the component called “Wallet”. This component will be used by all three actors, the issuer, the holder, and the verifier, and will be the same software solution for these actors. In Figure 9 the component diagram for the wallet is shown. The diagram focuses on the external interfaces provided and used by the wallet, while the internal interfaces between the modules are omitted to improve readability of the diagram. Also, the management interfaces, which are present for all modules within the wallet are combined into one *Management Interface* in the diagram.

[6] https://identity.foundation/presentation-exchange/spec/v2.0.0/#presentation-definition

[7] https://identity.foundation/presentation-exchange/spec/v2.0.0/#presentation-submission

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

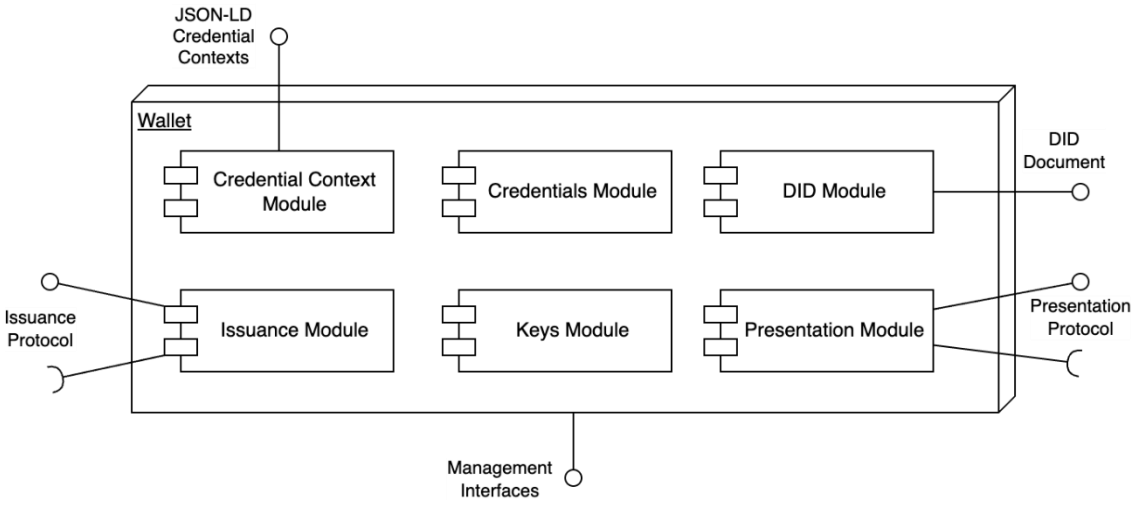


Figure 9: Wallet Component Diagram.

The responsibilities of the modules can be summarized as:

- **Credential Context Module:** responsible for managing and providing the relevant JSON-LD context definitions for the credentials this wallet will issue. For wallets that only use credentials but not issue credentials, this module will be empty.
- **Credentials Module:** responsible for managing the credentials issued by or issued to this wallet. This module is primarily focused on providing the credentials to the other modules (e.g. the presentation module and the issuance module).
- **DID Module:** responsible for managing the DID document of this wallet. Including the public key material and services related to this wallet instance. With a primary focus on making the relevant information publicly available. Also, resolving of DID identifiers to DID documents of other participants is offered as service to the other modules.
- **Issuance Module:** responsible for the issuance protocol, both for the issuer and holder. This module has tight relationships with the other modules (e.g. the credentials module, the keys module, and the credential context module).
- **Keys Module:** responsible for the private and public key material used by the wallet. Provides the public keys to the DID module and provides signature services to the other modules (e.g. the issuance module and presentation module)
- **Presentation Module:** responsible for the presentation protocol, both for the verifier and the holder.

The management interfaces are intended to be used together with the web-based User Interface (UI) (Figure 11), to provide the means for administrators to manage the wallet instance. Additionally, these management interfaces are also used by, for instance, the control

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

planes of the same participant to ensure the right credentials are presented required to establish trust.

3.6 Deployment view

The deployment view of the decentralized identity management is focused on the deployment of the Wallet. This deployment is assisted by a PostgreSQL database for persisting the state of the Wallet and to ensure all relevant information remains available for the Wallet. Also, an oAuth2.0 server is deployed to handle the internal authentication of administrative users that need to interact with the wallet. Figure 10 shows the deployment diagram for the deployments of the dataspace authority, and the deployments that participants may perform, which follow the same deployment scenario as the dataspace authority.

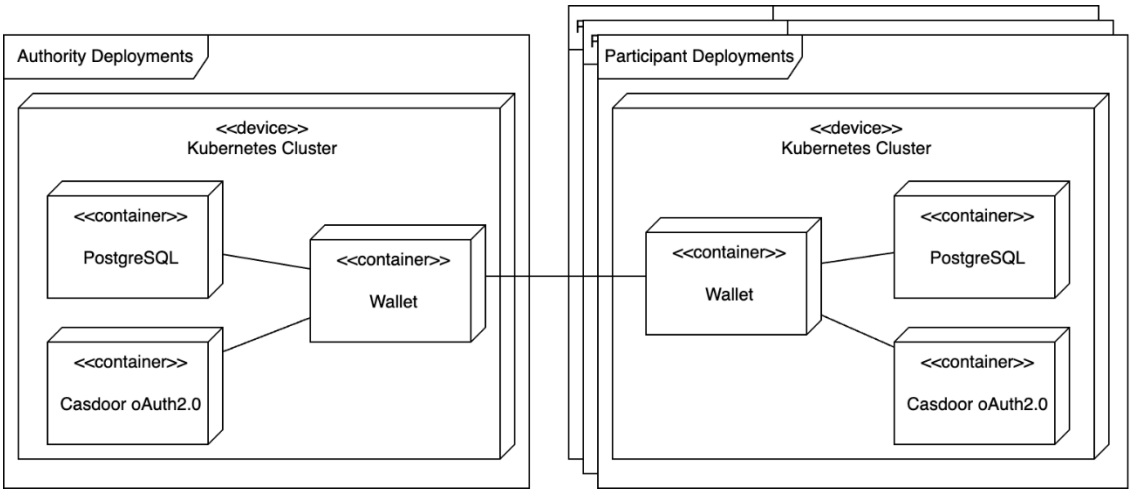


Figure 10: Wallet deployment diagram

The wallet for the dataspace authority is deployed under https://dsp.Enershare.dataspac.es/ and will remain available for the duration of the project.

Figure 11 is a screenshot of the UI of the Wallet, showing the overview of credentials the Wallet has issued.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

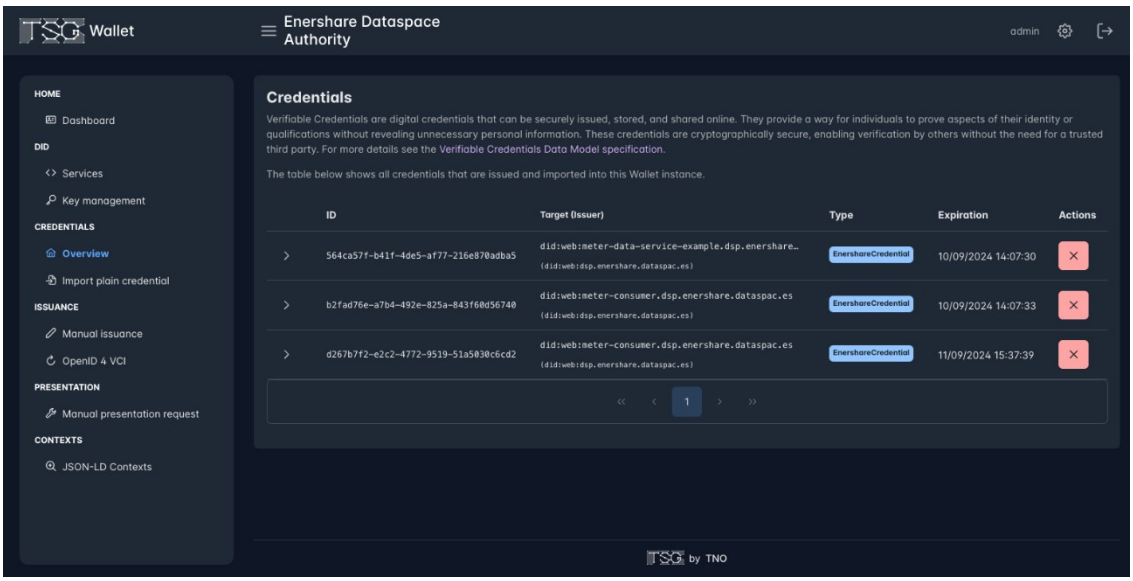


Figure 11: Screenshot of the wallet of the dataspace authority.

3.7 Identity management for end users (Keycloak - Marketplace)

The data space DAPS provides the first version of identity management for the data space. This implies the generation of identity certificates in X.509 format to authenticate and authorize *organizations* and their respective *connectors*, leaving humans in the loop without an authentication and authorization method that could directly represent them in the data space. The context and reference architecture of ENERSHARE considers tools and services, namely the Marketplace or the App Store, or the energy services developed in the context of WP6. To cater for this need, ENERSHARE merges two approaches:

1. the maintenance of the standard deployed authentication and authorization mechanisms in the data space for both *organizations* and *connectors*;
2. a complementary OAuth 2.0 and OpenID compliant Identity Provider for human users.

The solution that was adopted considers Keycloak[8] as an identity provisioning system to facilitate applications that require human interaction.

Keycloak provides several fine-grained authorization policy mechanisms, such as Attribute-based Access Control (ABAC), Role-based Access Control (RBAC), User-based Access Control (UBAC), among others. The former mechanisms are the ones more relevant to the scenario considered within ENERSHARE for the authentication of human users. These policy mechanisms allow to clear/not clear a given user to access resources that are considered restricted.

[8] https://www.keycloak.org/

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

The main components within the Keycloak architecture are the Policy Enforcer, i.e. the Policy Enforcement Point (PEP) that unlocks the set of permissions for the users or roles requesting it, the Policy Evaluation component, i.e. Policy Decision Point (PDP), that solves the decision process to either grant or refuse a set of permissions, establishing a disputable policy decision actuator and authorization services. Other components relate to the administration of the instance, such as the Policy Administration Point (PAP) or the resource server to serve all the inbound authentication and authorization protocols.

This solution does not preclude the already existing identity scheme. Thus, on the one hand, if a human user holds a data space connector that has onboarded the data space and does not require its direct, individual identity, it should consider the data space identity mechanism already in place. On the other hand, if a user requires to directly interface with the Marketplace, App Store or any other service from WP6 where its individual representation is considered, the complementary Keycloak system should be used to govern that identity lifecycle.

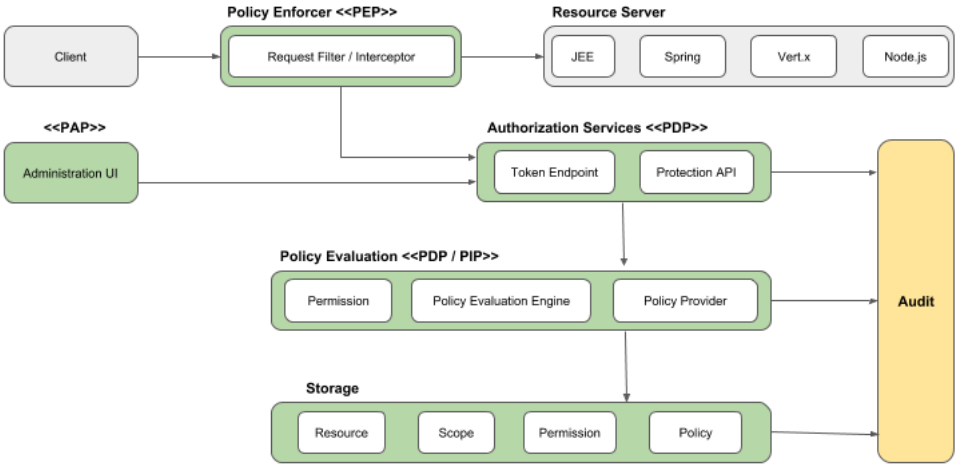


Figure 12: Keycloak official architecture[9].

Figure 13 depicts the described scenario, linking the App Store (and its internal components such as the docker image repository system i.e., Harbor), Marketplace, and all services that require this feature with the Keycloak instance. Please consider that this picture logically represents one Keycloak instance. Nonetheless, this instance may be configured using a federated configuration adopting several local authentication and authorization endpoints.

[9] As in https://www.keycloak.org/docs/25.0.0/authorization_services

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

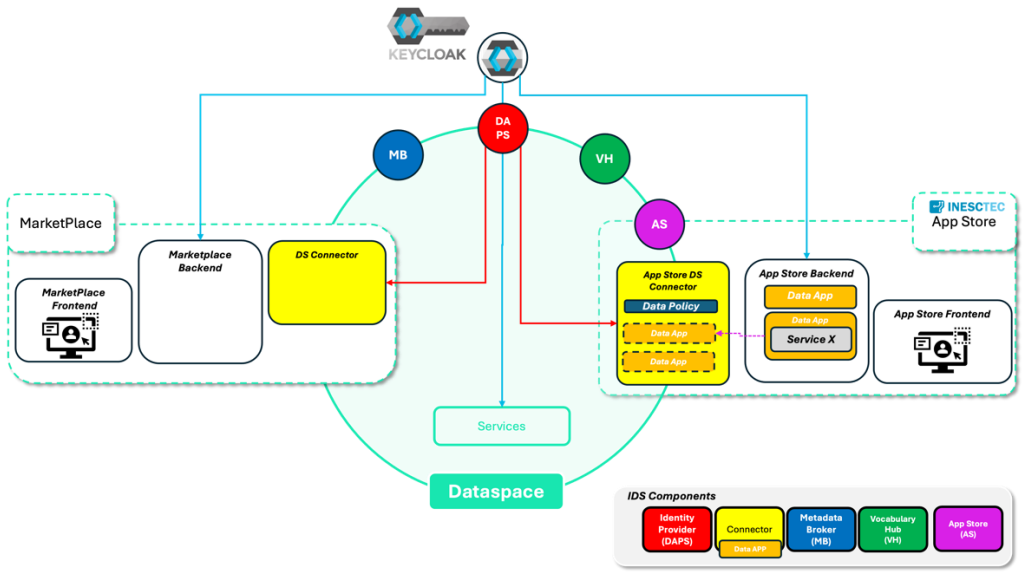


Figure 13: Keycloak IDP provisioning for components and services with human users.

The Keycloak instance deploys an authentication realm for ENERSHARE, where all user accounts and the necessary properties are kept. Currently, it holds the given names, surnames and email accounts as user identifiers. The sole purpose of this data collection is restricted to user experience in the stated platforms to authenticate and welcome the users. The configured realm could also be used to keep and association between the Keycloak user account and any (if present) data space connector- and organization ID that the user might have in place. Although human users in this setup do not directly participate in the data space with their individual identity, they may run services as part of a connector identity. This is relevant, for instance, in the App Store setup where a human user requires to authenticate to use this platform, but at the same time requires a data space identity to deploy Apps in the data space connector. In the specific case of the App Store Application, the internal docker image repository (via Harbor) also directly integrates with the Keycloak instance, establishing a OpenID Connect (OIDC) protocol link and allowing users to also be authenticated through the same Keycloak realm. In the scope of the App Store and the Marketplace, the Keycloak instance also provides means to grant roles to the user accounts.

All Keycloak client applications requiring authentication and authorization services integrate with existing provided client. Human users may refer to the App Store application or directly with the Keycloak instance to register one account.

This is possible via the authentication Uniform Resource Locator (URL), allowing to log in (Figure 14) or to create an account (Figure 15).

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

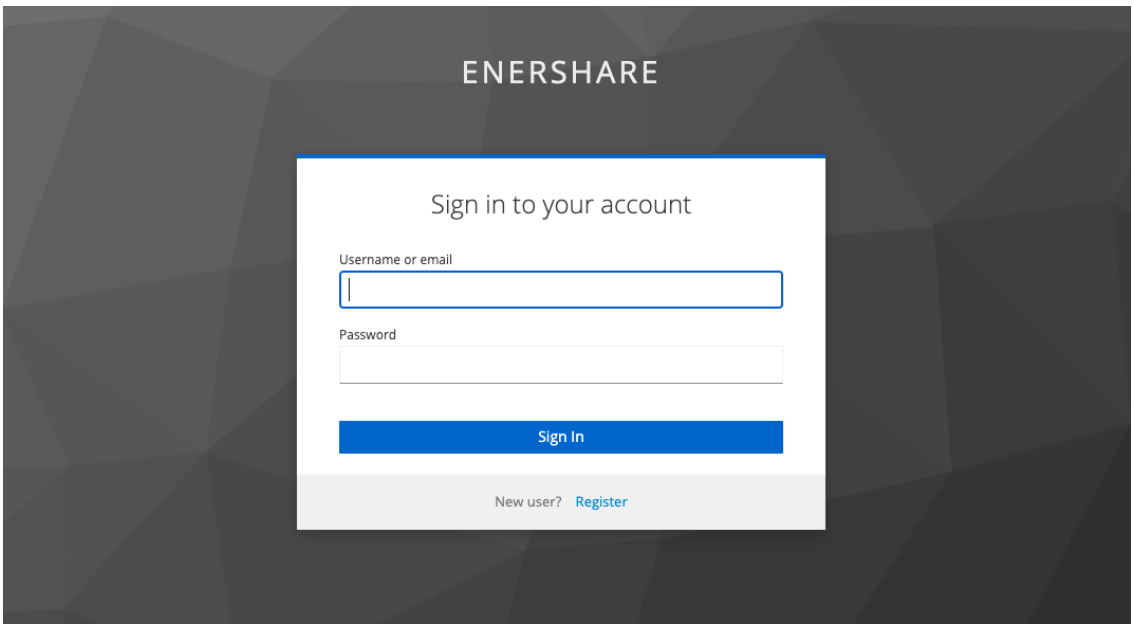


Figure 14: Keycloak authentication page[10].

[10] https://idp.haslab-dataspace.pt/realms/enershare/login-actions/authenticate

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

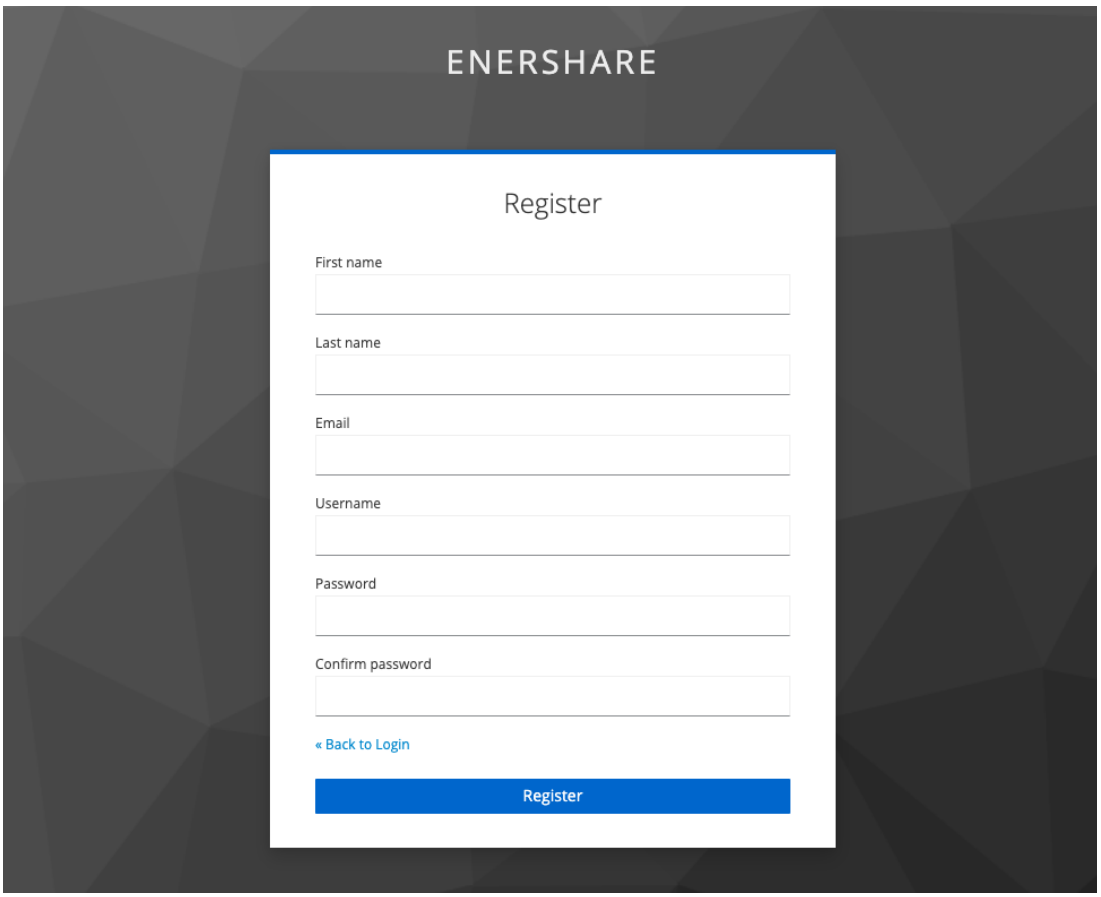


Figure 15: Keycloak register account page[11]

3.8 Conclusion

In recent developments concerning SUC1 (Onboarding), significant progress has been made to ensure seamless interoperability between the sister projects within the Energy Data Space project cluster. An alignment meeting was held between ENERSHARE and OMEGA-X[12], resulting in an agreement to use the OpenID for Verifiable Presentations (OID4VP) protocol for exchanging presentations of credentials between participants in ENERSHARE and OMEGA-X. Additionally, alignment on the issuance of credentials has resulted in the usage of the OpenID for Verifiable Credential Issuance (OID4VCI) protocol, although it is not strictly required for the interoperability between the projects aligning the issuance protocols will lead to harmonizing the processes.

[11] https://idp.haslab-dataspace.pt/realms/enershare/login-actions/registration

[12] https://omega-x.eu/

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Initially, the plan was to onboard a connector from ENERSHARE into OMEGA-X to obtain the verifiable credential. However, a key challenge arose due to differing setups: ENERSHARE follows a pre-authorized setup without a CA, while OMEGA-X employs an authorized setup with CA and DAPS.

To address this issue, a solution has been devised where each project will have its own issuer. Specifically, the connector from ENERSHARE will obtain the VC from its respective issuer and then present these credentials in OMEGA-X, where they are expected to be accepted. To facilitate this process, both projects will maintain a Gaia-X Digital Clearing House (GXDCH)[13] containing a list of accepted credential issuers.

It is important to note that this approach ensures that each project can maintain its identity and security protocols while still achieving interoperability. A Proof of Concept (PoC) for this solution is scheduled to be available in November for ENERSHARE MVP 3.0, marking a critical milestone in demonstrating the feasibility of cross-project credential acceptance.

Data Cellar[14], Synergies[15] and EDDIE[16] are currently assessing the feasibility of this approach, and some of them may also take part in the implementation of the PoC.

[13] https://gaia-x.eu/gxdch/

[14] https://datacellarproject.eu/

[15] https://synergies-project.eu/

[16] https://eddie.energy/

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

4 Usage control enforcement

In the initial phase of the project, the dataspace trust and sovereignty components were identified. Usage control and usage policy were discussed and evaluated, and all the currently available dataspace connectors were analyzed. Initially, the TRUE connector and TSG connector v1 were considered within the project scope, and future requirements from pilots were gathered accordingly. The pilot requirements, which mainly focused on the usage policy, were initially developed in the TRUE connector and then implemented in the TSG connector as part of T4.3.

The interoperability among the connectors is quite a challenge, which was realized during the development phases of the dataspace environment for the ENERSHARE project. The derived results for interoperability would require the connectors to implement the new Dataspace Protocol released earlier in 2024. Currently, TSG v2 has been able to develop the deployment in accordance with the new protocol and is still working on providing the usage control and policies initially developed in D4.2. At the time of writing, the activities on the TRUE connector for the alignment to the new DSP, which go even beyond the single scope of the project, have been started, and there are no plans yet for a release before the conclusion of the WP.

In January 2024, discussions with the TNO and ENG development teams highlighted that significant development work would be required to make the connectors compatible with the ENERSHARE dataspace, which goes beyond the scope of this WP. Following the decision, Fraunhofer IOSB-AST had to concentrate on the Eclipse Dataspace Connector (EDC) and open-source components, with a particular focus on researching and integrating usage control within the EDC. This work is essential for implementing the necessary usage control, policies, and a unified mechanism for policy enforcement and bridging the gap between the connectors. This chapter will explore the ongoing development efforts and how they contribute to achieving future interoperability among connectors.

4.1 Scenarios

The initial scope of the project included both the TRUE connector and the TSG connector as components of the Enershare dataspace. However, as the technical development progressed, and scenarios were detailed in deliverable D4.2, it was determined that the official connector for the project would be TSG. The TRUE connector will be further developed for deployment in the Enershare dataspace using a new data space protocol, which is beyond the scope of WP4. The TRUE connector remains a part of the project with some pilot implementations, and according to usage control for T4.3, it has already been integrated and was included in the previous release. The EDC, which was investigated in this deliverable, was included to research

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

the policy enforcement framework. This framework provides further conceptual implementation to help make connectors interoperable within the ENERSHARE Data Space. The scenario provides brief information on how to deploy and configure the EDC.

4.1.1 Deployment Eclipse Data Space Connector (EDC)

This scenario involves deploying two EDC connectors: one as a consumer and the other as a provider. The assets and policies are exchanged using contracts, and after contract negotiation, the asset becomes available to the consumer. The steps are as follows:

1. **Deploy Consumer EDC**: The consumer EDC is deployed using a docker-compose.yml file, configuring ports for the front-end, the EDC as the backend, and a Postgres database for the catalog. The consumer catalog is set up with the provider catalog URL to fetch available contracts for negotiation.
2. **Deploy Provider EDC**: The provider EDC is deployed using a docker-compose.yml file, configuring ports for the front-end, the EDC as the backend, and a Postgres database for the catalog. The provider catalog is set up with the consumer catalog URL to fetch available contracts for negotiation.
3. **Create Asset on Provider Connector:** An asset is created on the provider connector, which uses a UUID to map the data from the backend using its parameters.
4. **Define Policy:** A policy definition is created, such as a connector-restricted or time-restricted usage policy.
5. **Create Contracts**: Contracts are created using the assets and policies and then saved inside the provider connector catalog.
6. **Contract Negotiation Request**: The contract negotiation request is sent from the consumer side to the provider side for negotiation.
7. **Complete Contract Negotiation**: The provider accepts the contract, completing the negotiation. The contract is then transferred to the consumer, allowing the consumer to access the asset URL and data accordingly.

Outcome: This process facilitates interaction between the consumer and provider connectors with the data services deployed in the backend using the dataspace concept, incorporating usage control and usage policies.

4.2 Logical view

The EDC reference implementation is designed to adhere to several key responsibilities, including identity management, cataloging, discovery, contract management, data transfer, and service provision. As illustrated in Figure 16, the EDC integrates seamlessly into an IT infrastructure, working alongside systems responsible for monitoring and data provider or consumer. It fulfills multiple roles: it forwards requests while ensuring sovereign data

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

management, offers storage for persistent metadata and administrative data sources, and includes logical components that provide additional data space functionalities such as authentication, discovery, and usage control for data transfers.

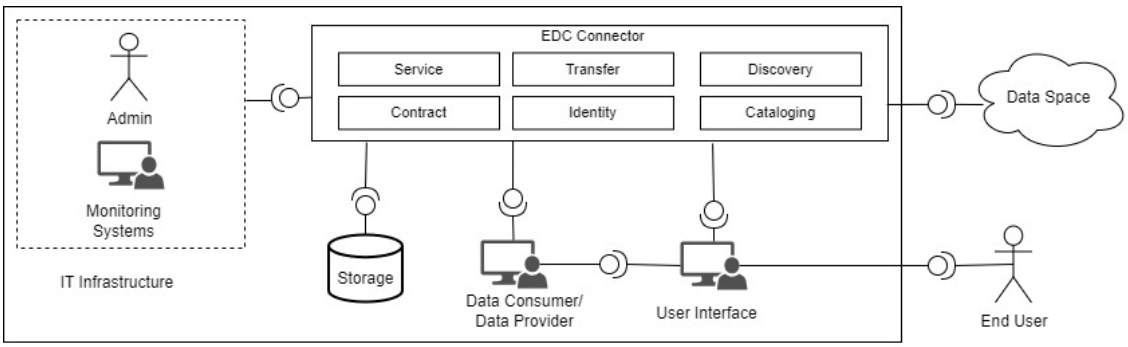


Figure 16: Consumer/Provider and System interaction in the EDC

4.3 Process view

In the beginning of the chapter, the deployment scenario of EDC was explained. Now, following that information, the process view is illustrated in Fig. 17 using a sequence diagram. The process follows these steps:

1. In the first step, the Data Provider EDC fetches the UUID from the UUID generator module, which creates and manages the mapping of the Data Asset.

2. This generated UUID is mapped to the Data Asset and parameters from the backend system, where the actual data is available for the end user.

3. Next, the step policies are created, i.e., time-restricted and connector-restricted, using the UI module inside the connector.

4. The contract is created with the details of Assets and policies, which are assigned to it. After that, these are saved in the Catalog on the Data Provider EDC. This contract list is available for the Data Consumer EDC connector to request.

5. The consumer requests the data since it already has the contract list from the provider EDC, which is also verified by the policy engine on the consumer side.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

6. Now the request is initiated from the consumer EDC. It is verified on the provider side, negotiation is completed, and then the review and signing of the contract are completed. Once that contract request is accepted, the next step is the transfer of the data.

7. From the Data Consumer connector, the EDC data request is accepted from the Provider EDC, and if everything is working, the requested data is initiated for the consumer.

8. In the final step, the data is transferred to the consumer side, which can be forwarded to their backend system using the provided URL. The history is maintained in the Provider EDC connector for the transfers that are completed successfully.

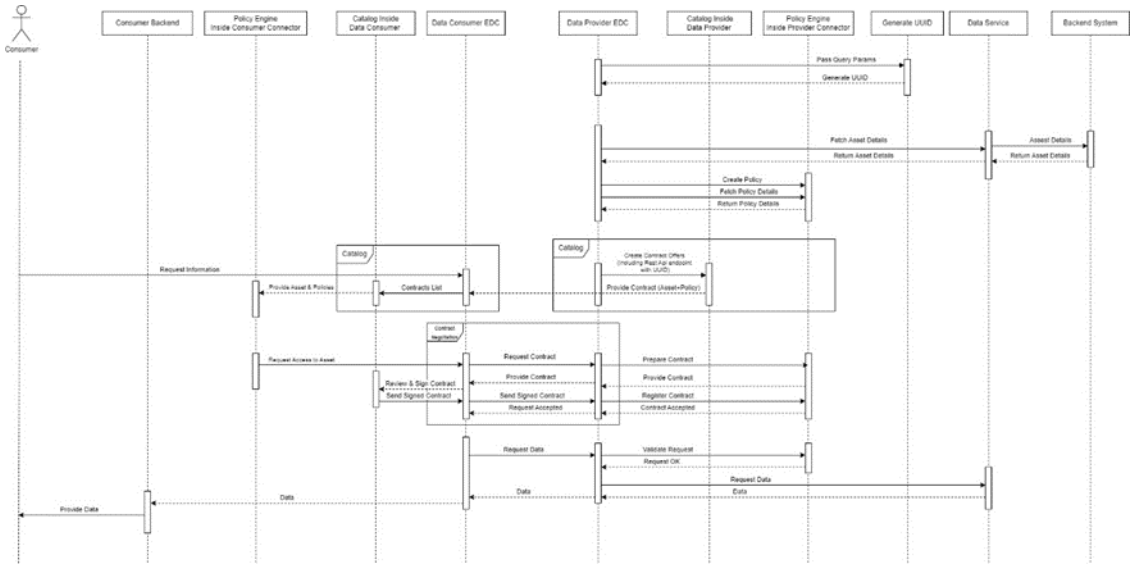


Figure 17: Sequence diagram for consumer/ provider using EDC

The explanation of the process currently works for using the extension built on EDC and the open-source implementation from Sovity[17]. Further development of the latest version of the EDC is still in progress and would require some additional changes to the current implementations.

[17] https://github.com/sovity/edc-ui

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

4.4 Development view

The interfaces within the EDC adhere to the IDS specifications, whereas all other interfaces accessible by data-consumer or data-provider connectors, graphical user interfaces, or human users (e.g., administrators) are built on widely accepted standards like HTTP 1.1 [1] and JSON [2]. Leveraging popular frameworks and programming languages facilitates straightforward integration, and the stateless architecture supports scalable development.

By utilizing Java, the EDC can operate within a container, providing versatile solutions for projects in various fields. This flexibility supports the implementation of straightforward data flows in diverse development and deployment environments while adhering to principles of sovereign data management. As shown in Figure 18 the functionalities demonstrate how to engage in a data space using an EDC. The illustration depicts a data space participant, who could be an individual or an organization, starting as a data provider (left side) and then becoming a data consumer (right side). Initially, the EDC is set up for the data space by integrating key components and assigning a specific identity. As a data provider, the EDC is then populated with metadata about the data intended for sharing. This data can be made available either through the component's IDS self-description or published in a central catalog system. Similarly, the EDC can function as a data consumer by searching the data space. Upon finding an appropriate offer, the participant can use the EDC Engine to negotiate a contract and download the data while ensuring policy compliance. Furthermore, the EDC can continually interact with a central catalog to log data requests and usages, and data flows can be augmented with different applications.

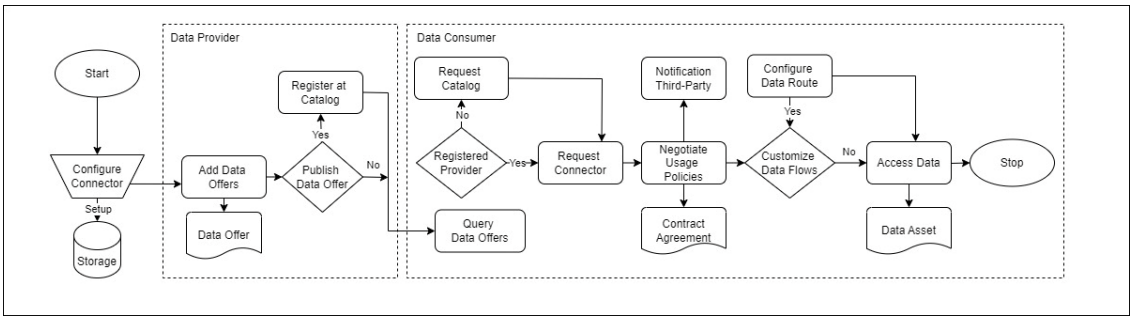


Figure 18: Process of data flow in EDC connector

The TSG, the official connector for the project, will support the previously drafted usage policy requirements. However, due to the implementation of decentralized identities, enforcing these usage policies will necessitate changes in future releases of the TSG v2 connector. Expressing a constraint in ODRL policy is only half the challenge; we also need code to evaluate it. The EDC registers evaluation functions with the policy engine to handle ODRL constraints. It employs the *MembershipCredentialEvaluationFunction* to verify membership credentials and the *DataAccessLevelFunction* to assess *DataAccess* level constraints based on the *DataProcessor*

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

credential. The EDC v0.8 policy definitions are more compliant with the new DSP, so adapting TSG v2 to these standards would enhance interoperability between the connectors and the ENERSHARE dataspace.

4.5 Deployment view

The deployment view of the EDC is focused on deployment of connectors in internal scenarios. The deployment uses a docker environment with the configuration for consumer and provider on different virtual machines. The Postgres database is configured inside the connectors for persistence of data and exchange of contracts. The production-ready deployment would require proxy configuration and security configuration from administrator would be required.

In the following Figure 19, the asset definition UI is shown in which the first UUID for time series is created with and then mapped accordingly with the asset in the next steps.

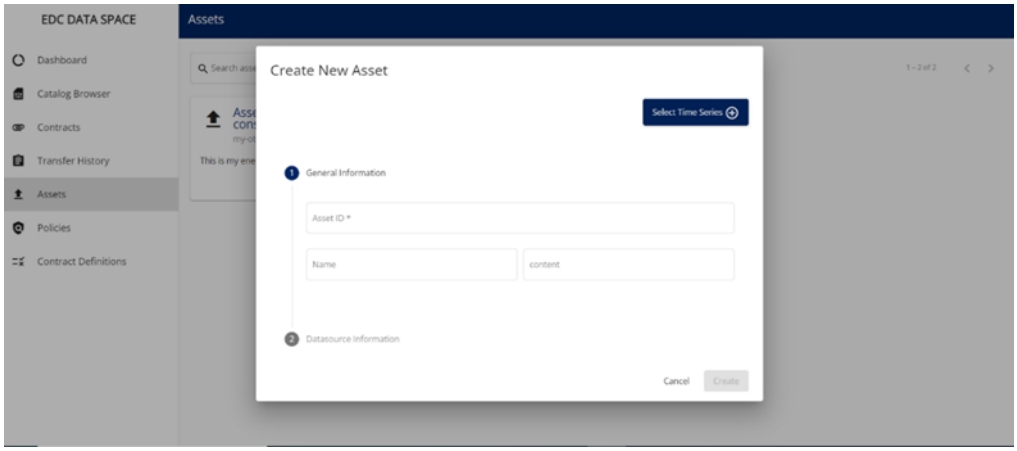


Figure 19: Create new asset definition

In Figure 20, the *create policy* UI shows the policy definition of the contracts, where the current policy is connector restricted in the example below.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

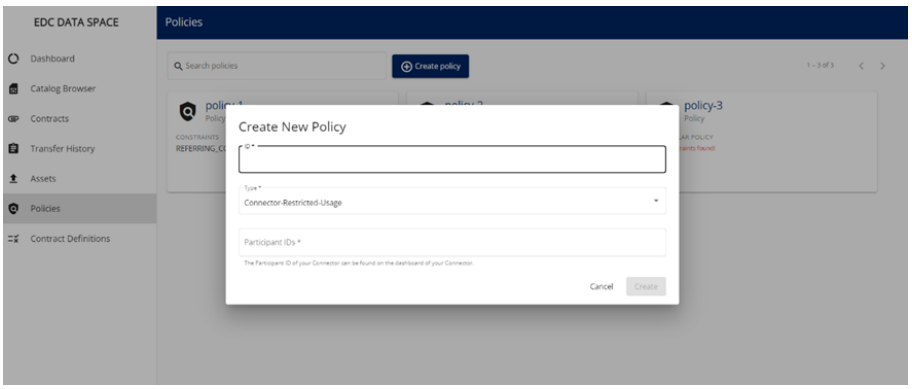


Figure 20: Create new policy definition

In Figure 21, the contract definition is shown in which the access policy, contract policy and assets information is selected, and the contract is created for future use.

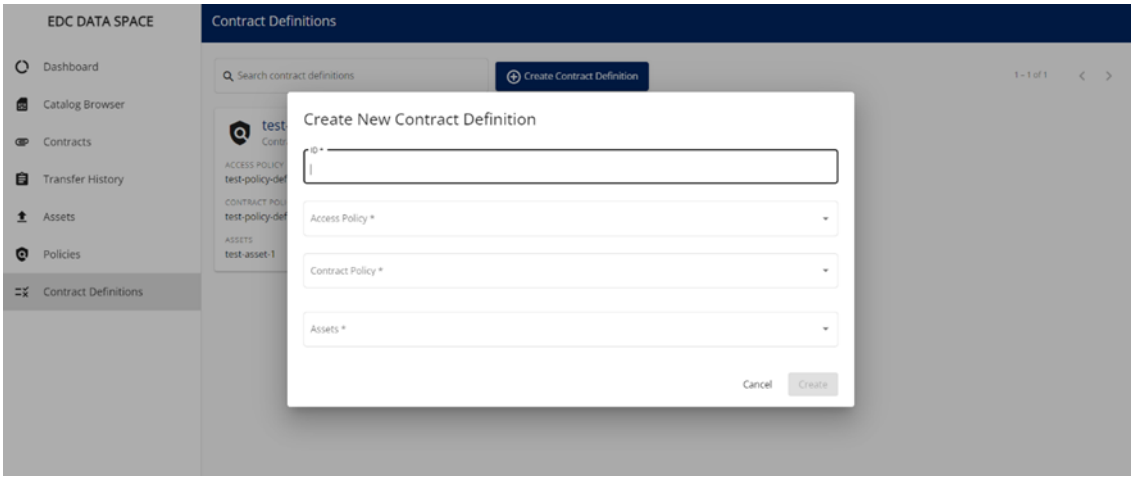


Figure 21: Create new contract definition

In Figure 22, the list of available contracts is shown and can be negotiated accordingly. The list of contracts consists of all the contracts available for the consumer side and the contracts already negotiated by the provider side.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

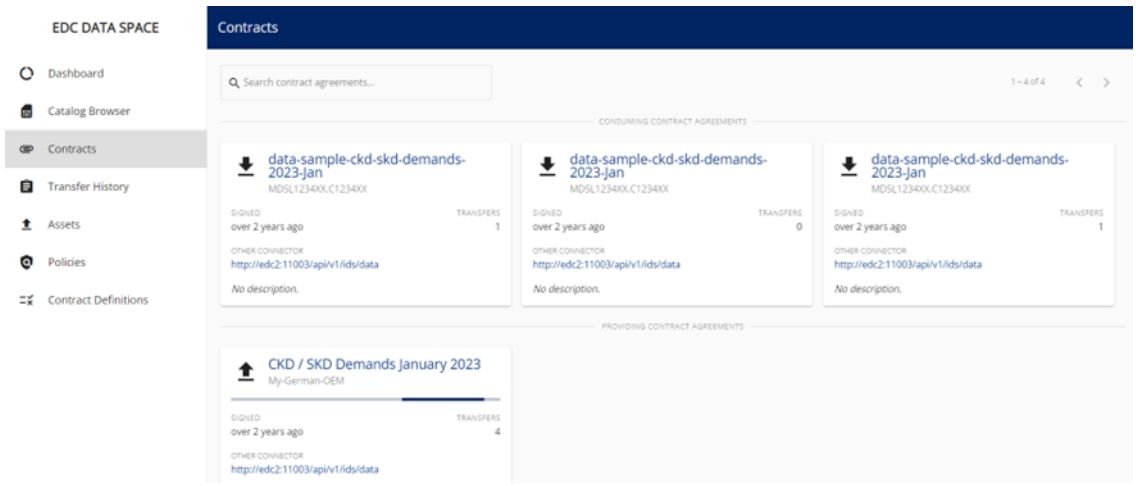


Figure 22: List of available contracts

In Figure 23, the history of the contracts currently transferred is shown. There are different states which show the state of the transfer was completed successfully or not.

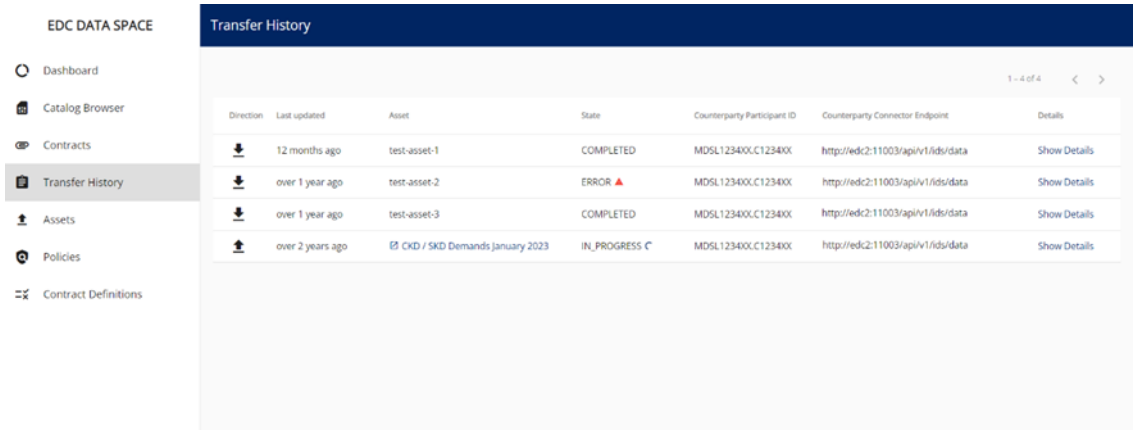


Figure 23: Transfer history for the contracts and assets

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

4.6 Conclusion

In the initial planning of the project and Task T4.3, the usage control and policy enforcement were focused on providing interoperability between data connectors, which would be important to the common data spaces. In the ENERSHARE project, the initial research, which was concluded in D4.1 and D4.2, aimed to support usage policies between the TRUE connector, TSG connector, and EDC connector. For this purpose, brief technical details were drafted previously to achieve the desired requirements.

In the previous version of TSG, the pilot requirements were discussed and implemented based on technical details. In recent months, the Dataspace Protocol has been released, which was implemented in a new version of the TSG connector. However, the support for previously drafted usage policy requirements and enforcement of these policies is still planned. This is because the policies are likely to be structured differently, due to the usage of decentralized identity management in the Dataspace Protocol, and other concepts that will change the internal behavior of policies.

Our work is focused on enhancing interoperability between data connectors, a crucial factor for the success of future projects within common data spaces. Specifically, we have concentrated on usage control and policy enforcement within the EDC by developing an EDC Extension that ensures consistent application of usage policies across various connectors. This integration also facilitates automatic contract negotiation through the front end, thereby significantly streamlining the data exchange process. Additionally, we have emphasized the development of a usage control mechanism that enforces policies using the ODRL format. While this mechanism is currently undergoing internal testing, it will soon be validated in real-world scenarios. Although full integration with the new Dataspace Protocol and interoperability with the latest version of the TSG connector are beyond the current project's scope, these aspects will be supported in future developments. By establishing common exchange standards, compatible policies, and effective enforcement mechanisms, our work provides a solid foundation for seamless interoperability between connectors, which is essential for the continued evolution of data space technologies.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

5 Usage policy notarization on the blockchain

Since its initial early prototype release, the Blockchain Notarization Module has evolved into a complete tool, not only linked to the Usage Control, already described in the previous section, but appliable at different levels of the entire ENERSHARE stack. As a matter of fact, the tool has been used also by other WP components (e.g., the Marketplace) and in different pilots, exploiting its versatility. The module consists of a smart contract, the Proof of Existence (PoE), in two different versions, and a set of API to improve its usability and adoption in a wider number of use cases.

In the previous deliverables D4.1 and D4.2 the state of the art, requirements and functionalities of the blockchain module have been described, along with the first version of the PoE (v5) released as demo and early version of the tool. In the current document the final version of the PoE (v9) is presented, evolving the functionalities of the v7 of the contract (presented in D4.2) which can now be considered as deprecated. The PoE_v5 was instead maintained for its simplicity and usefulness in some cases, refined and finalized as well. In brief, PoE_v9 adds nullification and history of changes to the notarization and verification functionalities already implemented by PoE_v5.

5.1 Scenarios

5.1.1 Notarize, nullify and verify usage policy definitions and documents

This scenario involves in the notarization, nullification, and verification of a usage policy definition as well as a generic document into the blockchain (Figure 24 and Figure 25). The term “document” can then represent the usage control policy definition as well as any other text. In this scenario, a document provider can decide to notarize just the content of a document or to also provide a description of the action performed, containing, for example, the reason of the action. The same document provider can also decide to nullify an already notarized information, adding, also in this case, more information related to the nullification action. Other users, like a document verifier, can decide in any time to verify the notarization status of a specific document providing the document itself.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

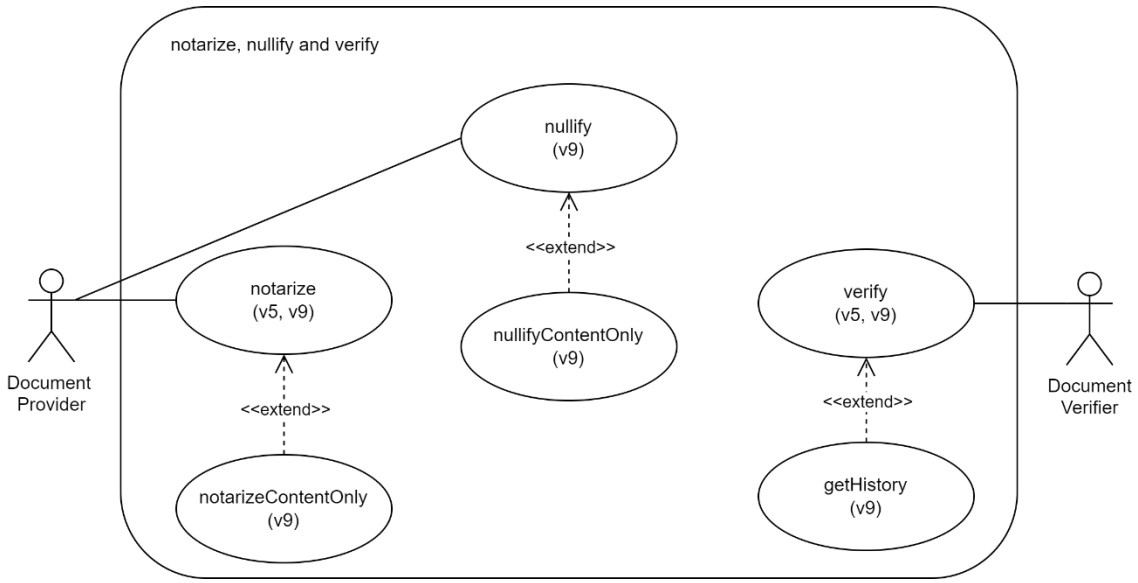


Figure 24: Use case for notarize, nullify, and verify documents

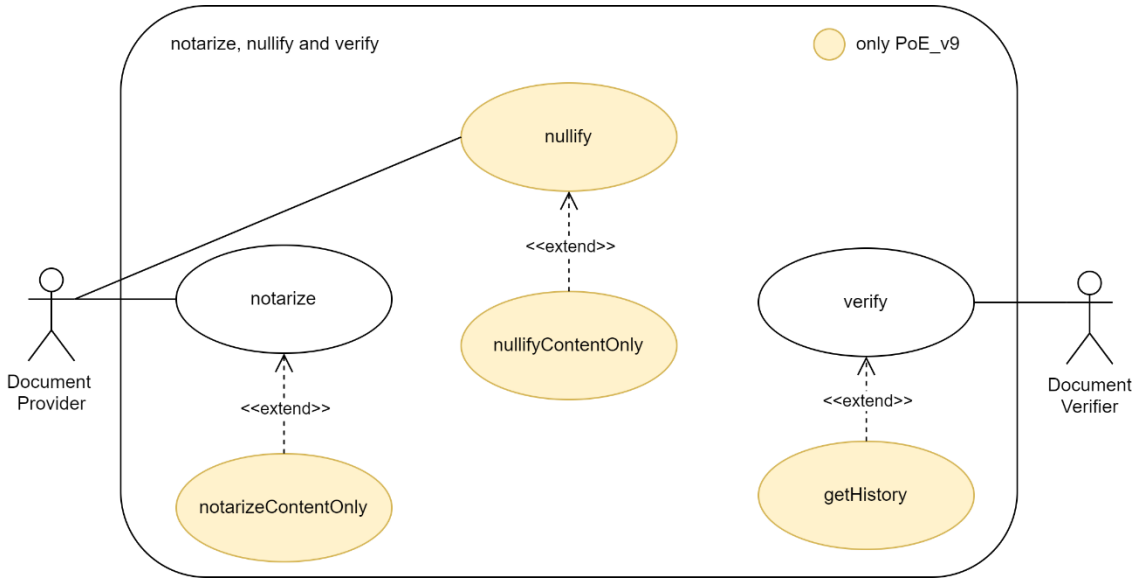


Figure 25: Use case for notarize, nullify, and verify documents

5.1.2 Auctions management in the ENERSHARE Marketplace

The scenario depicted in Figure 26 is related to another utilization of the blockchain module, adapted for the auctions section of the ENERSHARE Tokenized Marketplace (WP5). This use case involves the managing of auctions to exchange resources with cross domain services or

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

other resources. The market participant as an auction promoter creates an auction selecting one of his resources to exchange and sets the start and the end of the auction. The creation of the auction with the reference to the resource ID and to the start and end of the auction are notarized into the blockchain using the PoE_v5 smart contract. During the period when the auction is open, others user as auction bidders can propose one of their resources or a cross domain service as a bid to the selected auction. Each bid is notarized in the blockchain using the PoE_v5 smart contract by indicating some information such as the resources in exchange and the timestamp of the bid. Finally, at the end of the auction, the auction promoter can visualize all the bids and select the winning one. The choice made is notarized in the blockchain using PoE_v5 smart contract together with the reference to the winner bid and the timestamp of the notarization.

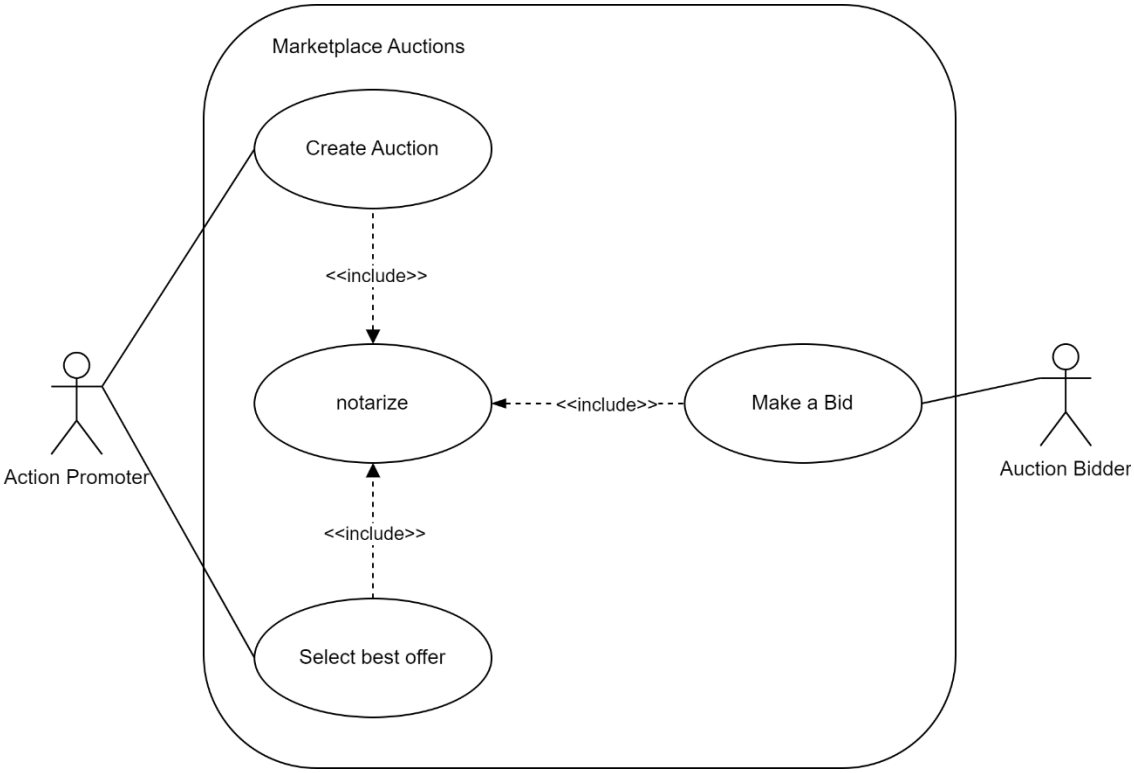


Figure 26: Use case for the auctions management in the ENERSHARE Marketplace (WP5)

5.2 Logical view

Figure 27 shows the class diagrams for PoE_v5 and PoE_v9. The methods are divided into internal, external and public. The internal methods are utility functions callable only within the

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

contract itself and any derived contracts, the external methods can be called only outside the contract itself and the public ones can be called from both inside and outside the contract.

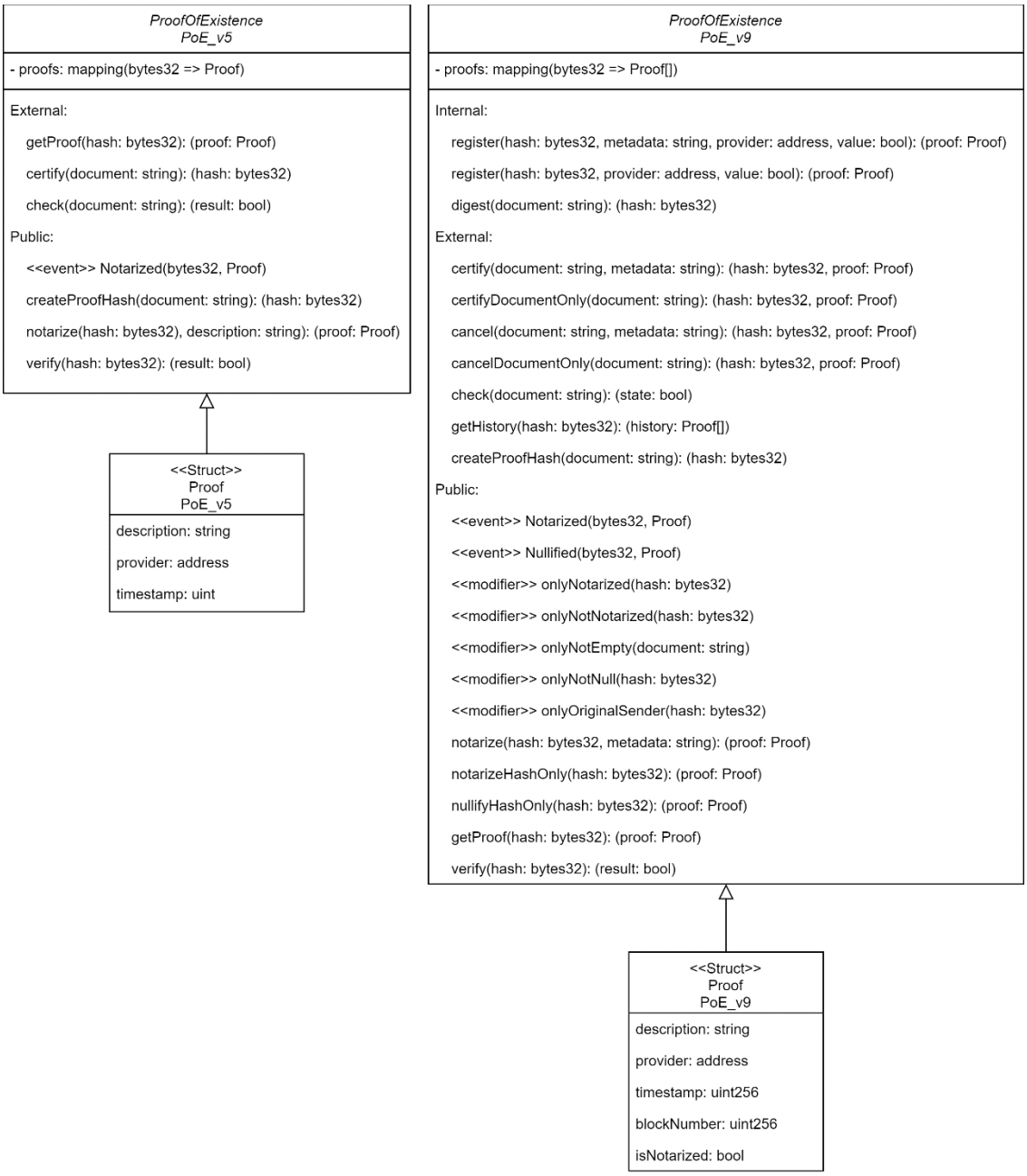


Figure 27: PoE_v5 and PoE_v9 class diagrams.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

5.3 Process view

This section contains the sequence diagrams for the main operations that can be performed by the PoE component.

Figure 28 and Figure 29 show the sequence diagrams for PoE_v5 and PoE_v9 respectively for the notarization of a document. In the first case the content of the document is notarized together with a description, in the second case without it. In both sequence diagrams, the APIs server handles, on behalf of the user, the generation of the hash of the document. The PoE then performs the transaction in the blockchain and returns the notarized hash by emitting the corresponding asynchronous event.

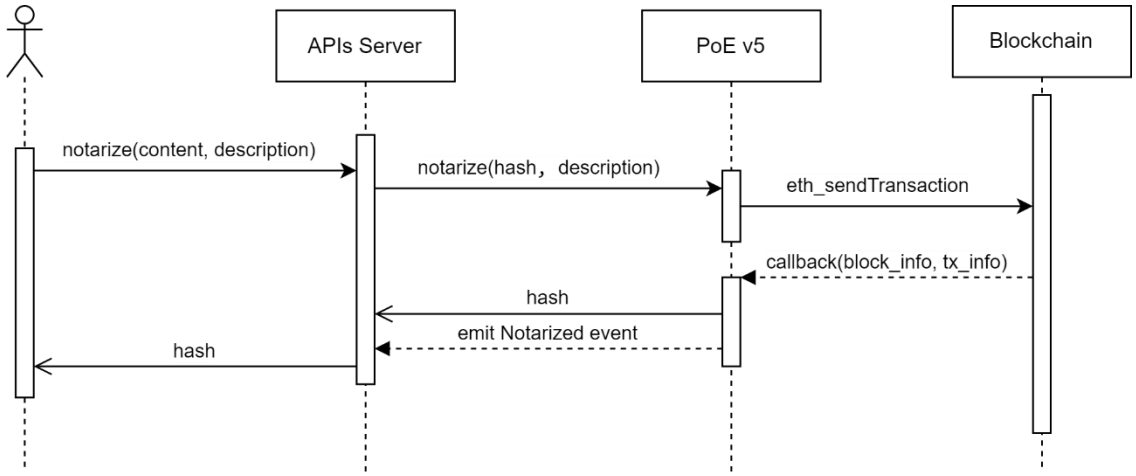


Figure 28: Notarize sequence diagram for PoE_v5

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

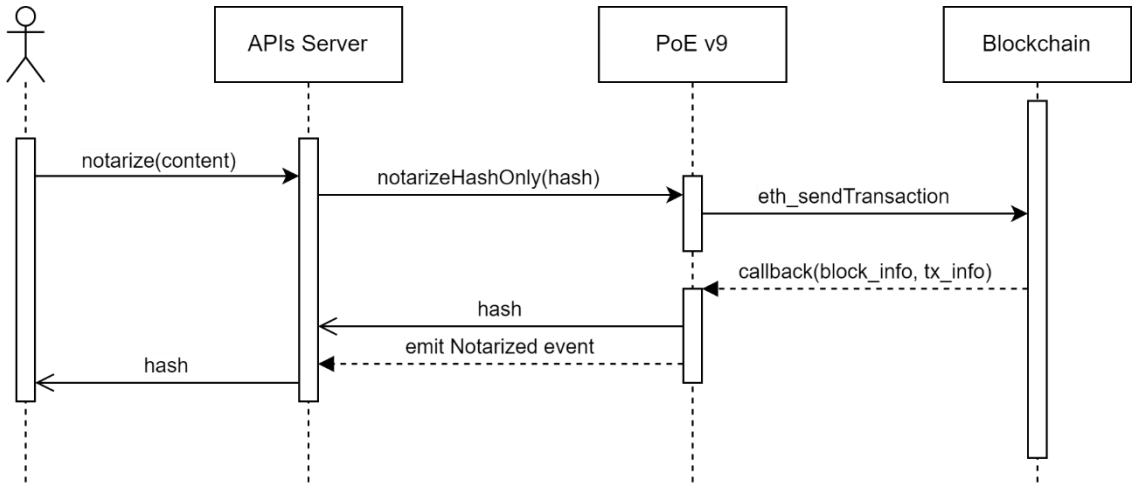


Figure 29: Notarize content only sequence diagram for PoE_v9

In the case of document verification, the user directly sends the contents of the document to the APIs Server, which is responsible for calculating the hash of the document and invoking the smart contract verification method. This operation is a read operation in the blockchain and does not require any payment.

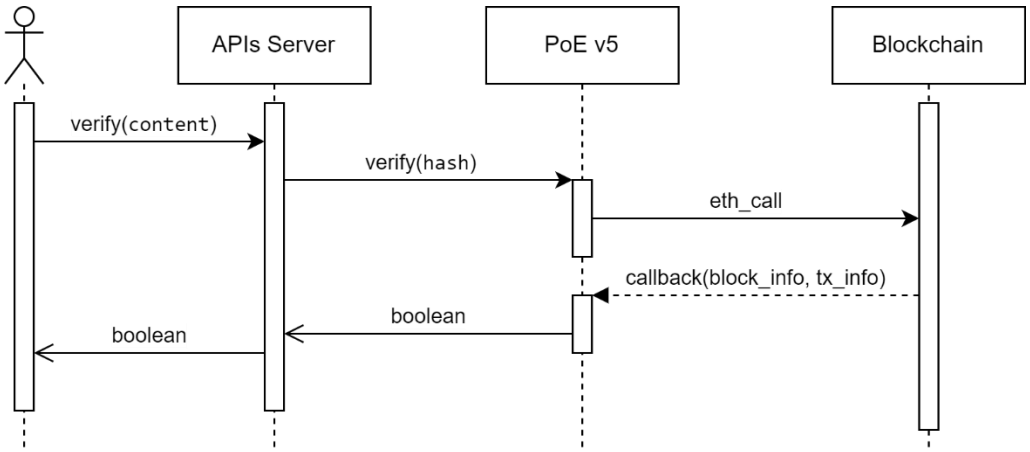


Figure 30: Verify sequence diagram for PoE_v5

Similarly to the case of notarization, Figure 31 and Figure 32 show the same nullification operation with and without a description. The user sends to the APIs Server the content of the document to nullify and, optionally, additional information. The APIs Server calculates the hash

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

of the document to nullify and invokes the corresponding method of the PoE and the smart contract performs the transaction on the blockchain. Then the PoE returns the hash of the nullified document to the user and emits the related event.

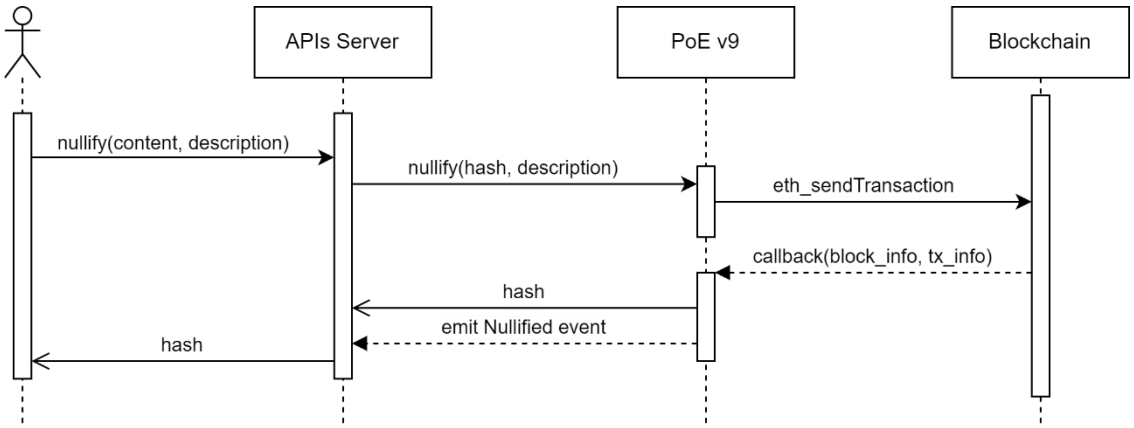


Figure 31: Nullify sequence diagram for PoE_v9

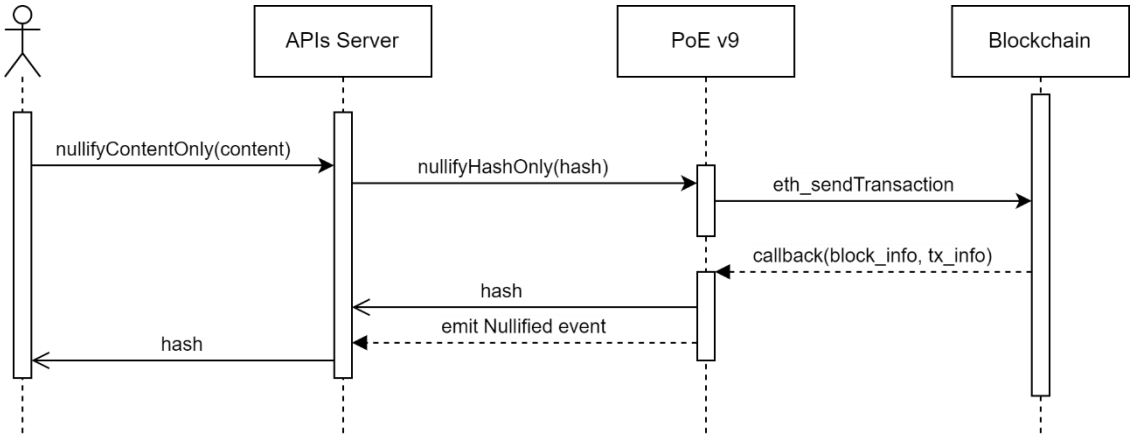


Figure 32: Nullify content only sequence diagram for PoE_v9

Figure 33 shows the call method for the retrieve of the proof of a document notarized. The user sends the content of the document for which they want to retrieve the proof of notarization. The APIs Server calculates the hash and invokes the corresponding method of the PoE. Finally, the PoE returns the proof, retrieved from the blockchain, containing the description (if any), the address of the provider, the timestamp of the notarization, the block number of the blockchain and if it is a valid notarization or not.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

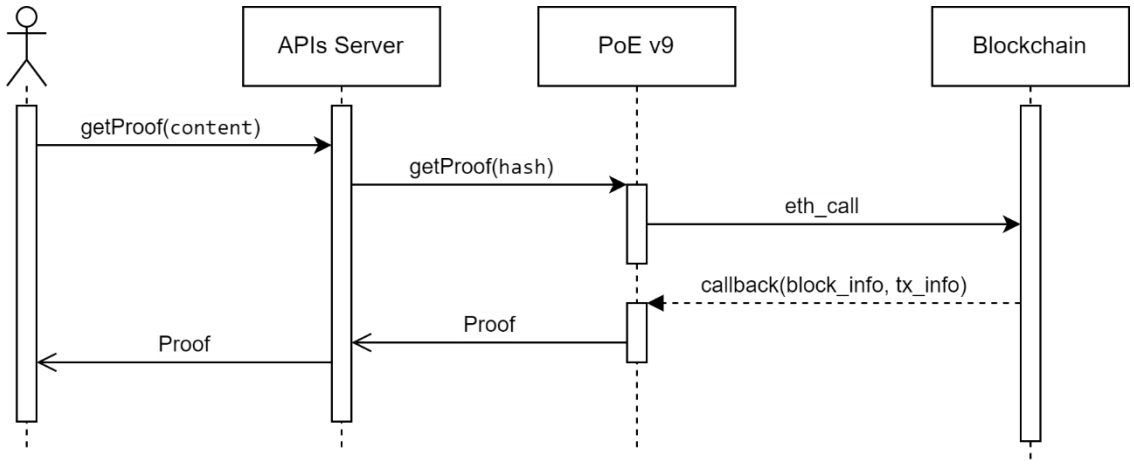


Figure 33: Get proof sequence diagram for PoE_v9

Figure 34 shows the operations required to retrieve the history of a certain notarized document. The user sends the content to the APIs Server which calculates the hash and invoke the getHistory method of the PoE, then the PoE retrieves from the blockchain the list of all Proof related to that hash and returns them to the user.

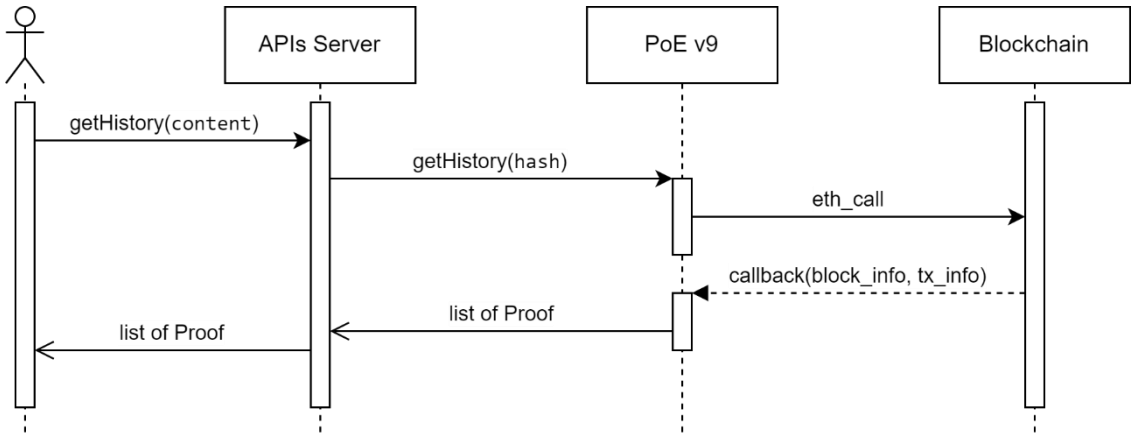


Figure 34: Get history sequence diagram for PoE_v9

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

5.4 Development view

Figure 35 and Figure 36 show the component diagrams for PoE_v5 and PoE_v9. In both cases, components have been designed for each macro functionality offered by smart contracts so as to facilitate its use and make it as transparent as possible.

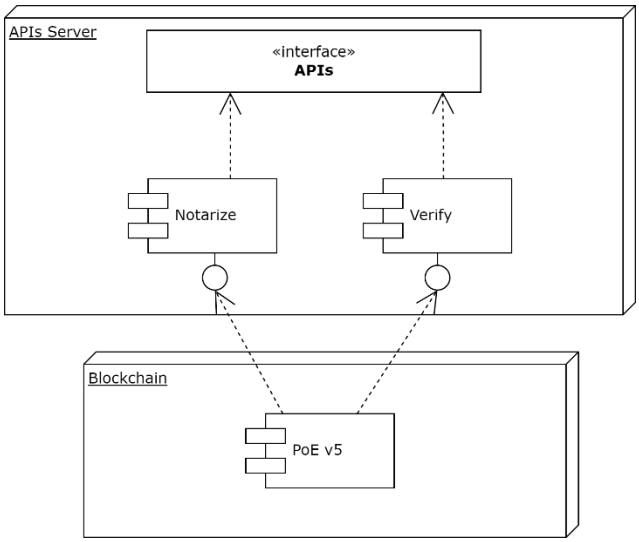


Figure 35: PoE_v5 component diagram

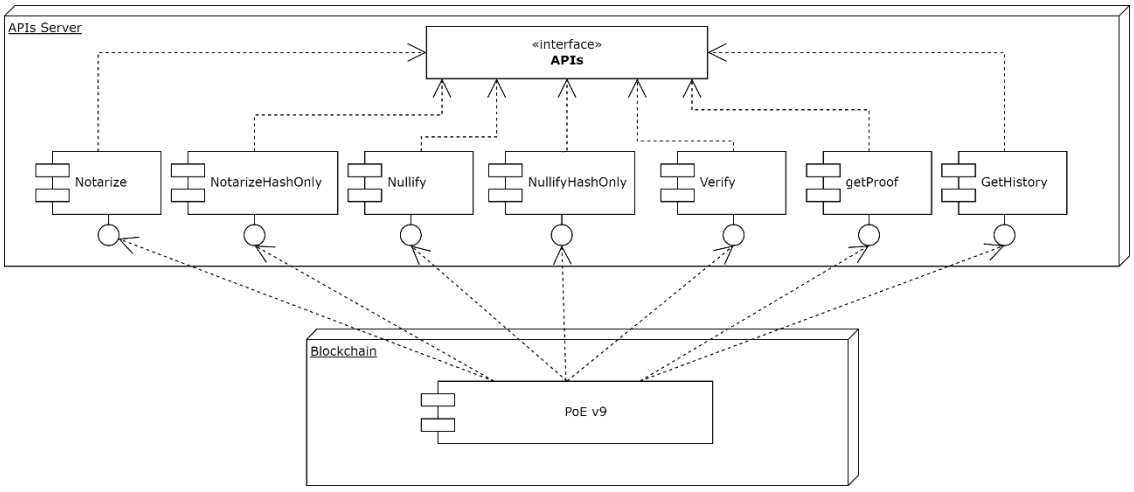


Figure 36: PoE_v9 component diagram

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Together with the API, a Swagger interface has been configured to test and use the notarization tool, serving also as guide and specifics for the programmers (Figure 37). The following figure shows the Swagger section for the fundamental notarize and verify APIs.

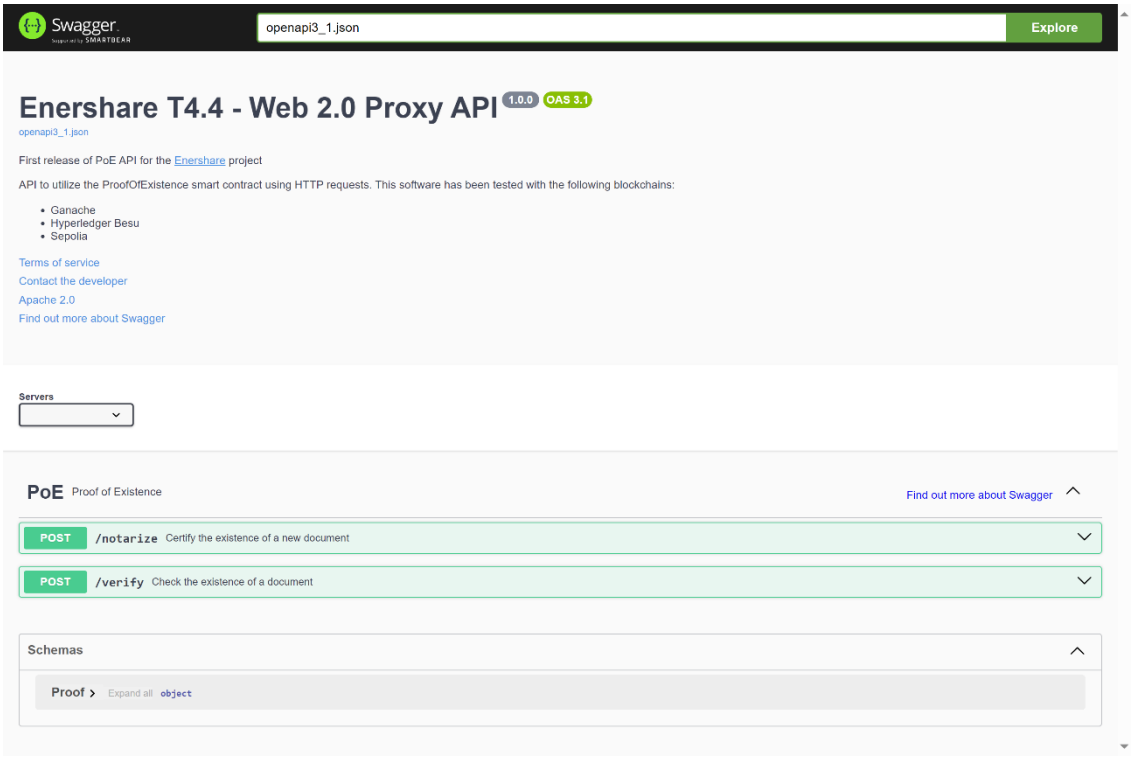


Figure 37: Swagger interfaces for the PoE APIs Server.

5.5 Conclusion

Initially conceived as support for Task 4.3 outcomes about usage control, the Blockchain Notarization Module has evolved into a complete versatile tool, appliable at different levels of the ENERSHARE stack. The module consists of a smart contract, the PoE, developed in Solidity and deployed into an Ethereum-based blockchain, at its core. The Sepolia testnet has been used for tests and is currently in use, while a Dockerized stand-alone distribution has been developed and released, as well, for rapid private chain deployment.

Among the different versions developed and released of the PoE, two of them, v5 and v9, emerged as the most appropriate for the different use cases: PoE v5 implements a classic notarization service, while PoE v9 adds nullification and historicization of the proof states, to permit a wider range of usages.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

The smart contract has been provided with a standard APIs layer, using a classic Web 2.0 based approach, to ease the integration of Web 3.0 based techniques to a wider range of possible adopters. The API layer, implemented as REST services, is an attempt to find the best trade-off between usability/user-friendliness and the advantages granted by the blockchain technology, which is still complex and some-how experimental, compared to the more consolidated traditional ones. User guides and Swagger interfaces have been released, as well.

Beside the Usage Control tool, the Blockchain Notarization Module has been distributed to some ENERSHARE pilots, and in particular in ENERSHARE Pilot 3, to support anonymization and certification of specific data sets, and in ENERSHARE Pilot 5, to enable the trade of different types of assets in an automated, decentralized, and flexible way. The PoE is also used inside the ENERSHARE Marketplace developed in WP5, in the Auction section, for the notarization of the openings, the bids, and closing with the winning one. So, indirectly, the Blockchain Notarization Module will be deployed and used also in ENERSHARE Pilot 4 and Pilot 7, where the usage and test of the Marketplace is envisaged.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

6 Remote attestation for full stack integrity

Since ENERSHARE deliverable D4.2, advancements have been made in the architecture surrounding remote attestation by leveraging Confidential Containers[18]. The Confidential Containers (CoCo) project is an open-source initiative in the Cloud Native Computing Foundation, which supports different types of trusted execution environments, like Intel SGX, Intel TDX, AMD SEV, and Arm TrustZone. The goal of CoCo is to standardize confidential computing for containers and simplify the usage of confidential computing in Kubernetes.

Within data spaces, remote attestation of workloads is particularly of interest to allow increased trust in the applications using data that is deployed by a remote entity. In addition to that, it is important to minimize the Trusted Compute Base (TCB), since this covers all aspects important to the confidentiality of workloads, including hardware, firmware, and software. To increase usability and applicability of remote attestation, constraints on the workloads should be minimized as these often limit the usable programming languages and require specific compilation into specifically crafted enclaves.

The current section describes the usage of CoCo in combination with AMD SEV-SNP to demonstrate remote attestation. It allows deployment of unmodified applications to be executed, while keeping the TCB as small as possible. By creating sandboxes for workloads in Kubernetes -using Kata Containers[19]- each workload is executed in a Utility Virtual Machine (UVM), shown in Figure 38 as a schematic representation. Note that only the green elements of the UVM and the workload pod itself are within the TCB (Figure 38). Everything outside of this UVM, like the Kubernetes components, but also untrusted workload pods on the same machine, are outside of the TCB. This results in a situation where an external entity only needs to be able to attest the resources inside the UVM to ensure the trustworthiness of the workload pod.

[18] https://confidentialcontainers.org/

[19] https://katacontainers.io/

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

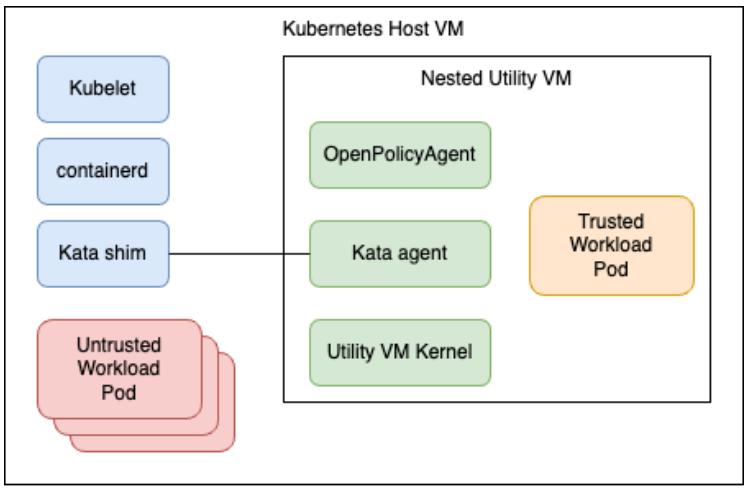


Figure 38: Nested utility virtual machine

6.1 Scenarios

The scenarios for remote attestation can be split up into three phases:

1. preparation phase
2. remote attestation phase
3. usage phase.

In the preparation phase (1.), the foundation is created to trust the application running inside the trusted workload pod. In the remote attestation phase (2.), the running trusted workload pod is attested to ensure that the runtime environment of the workload pod is in a trusted state. Additionally, keys are exchanged in this phase to allow data to be exchanged securely with the workload. Finally, in the usage phase (3.), the actual workload is executed by either transferring data towards the workload or by exchanging information from within the workload to another application. These phases are described in more detail in the subsequent paragraphs.

The above mentioned scenarios assume two roles: first, the *verifier* that wants to ensure the remote workload is trustworthy, and, second, the *prover* that wants to prove that its workload is trustworthy.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

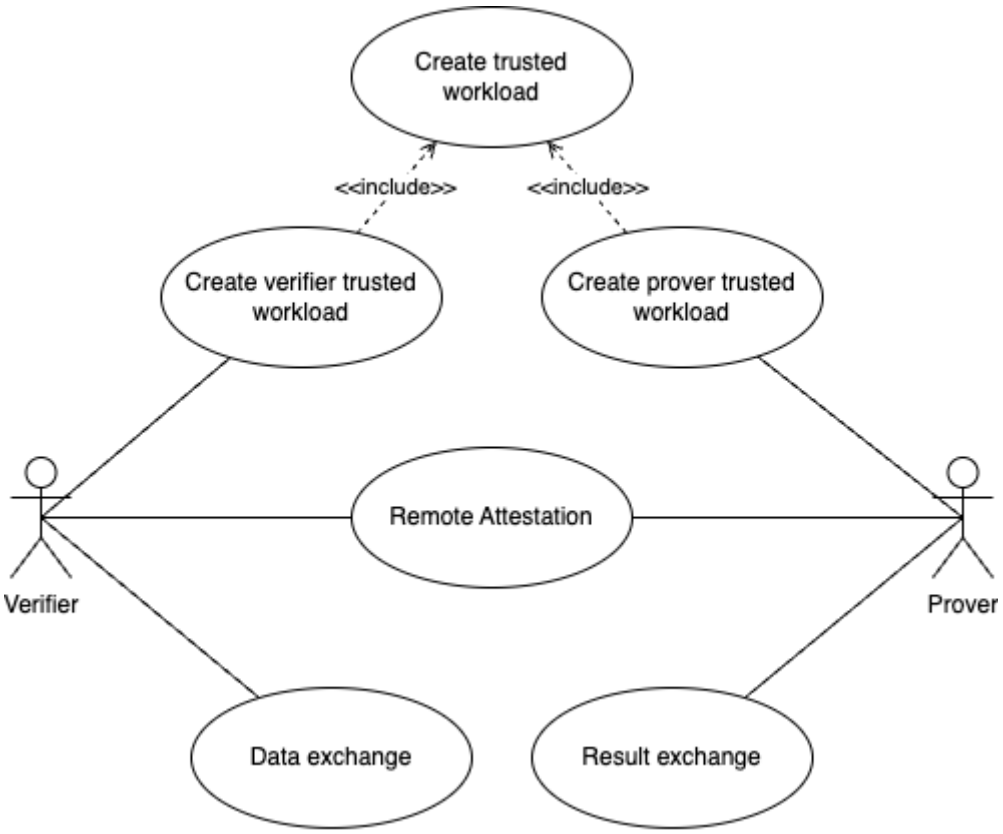


Figure 39: Remote attestation use case diagram

6.1.1 Preparation phase

This section describes the first of the three stages of remote attestation., the preparations phase., consisting of the creation of the trusted workload, of its verifier, and of its prover. These scenarios are described in detail below.

6.1.1.1 Create trusted workload

The first scenario involves creating a trusted workload by developing an application, defining its deployment configuration in the form of a Kubernetes pod, and establishing a Kata policy to ensure the security and integrity of the workload, using the following steps:

1. **Application development**: Develop the application code using preferred programming languages and frameworks. Containerize the application and publish the application in a container registry accessible by both verifier and prover.
2. **Kubernetes Pod configuration:** Create the Kubernetes Pod configuration metadata to allow consistent deployment of the application in Kubernetes clusters, including a runtime class name that is configured to run with Kata CoCo isolation.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

3. **Kata Agent Policy creation**: Create the Kata Agent Policy using the Rego policy language, defining the allowed actions. Including, in most situations, checks on the container image layers and restrictions on execution of commands and access to files/logs from outside of the pod.

Outcome: a trusted workload is created and is deployable in a Kubernetes cluster, running with enhanced security provided by Kata Containers and CoCo.

6.1.1.2 Create verifier trusted workload

This scenario includes the *Create trusted workload* scenario. The verifier is executing the *Create trusted workload* scenario and provides the application, pod configuration, and the Kata agent policy to the prover. Since the verifier has created the application, there is no need to verify its contents by a third-party or even the prover. Especially when the workload sources contain intellectual property the verifier doesn't want to share with the prover.

The scenario involves the following steps:

1. **Create trusted workload.** See above.
2. **Share workload configuration:** Exchange (a reference) to the workload container image, the Kubernetes pod configuration, and the kata agent policy with the verifier. This can be done outside of the data space to allow for deployment of the workload statically. Or, within the data space by exchanging the resources via a data plane that supports orchestration of the workload in a dynamic manner.
3. **Deploy trusted workload:** The prover deploys the workload in a pod sandbox environment using Kata containers with AMD SEV-SNP enabled.

Outcome: a trusted workload is created by the verifier is exchanged with the prover and the workload is deployed as a confidential container.

6.1.1.3 Create prover trusted workload

This scenario includes the *Create trusted workload* scenario, as described above, executed by the prover. Depending on who verifiers the application, the prover exchanges at least the Kata agent policy with the verifier and, optionally, also the application sources are shared with the verifier. The steps in this scenario are:

1. **Create trusted workload.** See above.
2. **Deploy trusted workload.** The prover deploys the workload in a pod sandbox environment using Kata containers with AMD SEV-SNP enabled.
3. **Share Kata agent policy and related workload information.** Exchange the Kata agent policy from the prover to the verifier. This can be done outside of the data space. Or, within the dataspace by including the agent policy in the contract negotiation or

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

transfer process. Additional information the verifier needs to ensure trust in the workload is exchanged alongside the Kata agent policy.

Outcome: a trusted workload created by the prover is deployed as a confidential container. The relevant information of the trusted workflow is exchanged with the verifier to allow the verifier to trust the workload.

6.1.2 Remote attestation phase

This section describes the second of the three stages of remote attestation., the actual remote attestation.

6.1.2.1 Remote Attestation

This scenario outlines the process of remotely attesting the workload of the prover by the verifier. During the attestation, a shared key is generated that can be used for encryption of data between the verifier and prover. It includes the following steps:

1. **Request certificate chain.** Request the certificate chain used for the attestation of the TCB. For AMD SEV-SNP, this will contain the AMD Root Key (ARK), the AMD Signing Key (ASK), and the Versioned Chip Endorsement Key (VCEK).
2. **Validate certificate chain.** The verifier validates the signatures of the certificate and checks the ARK with the publicly available ARK published by AMD.
3. **Prepare Diffie-Hellman key exchange.** Both verifier and prover will prepare a public-private key pair for the Diffie-Hellman key exchange. Based on a well-known curve, e.g. the NIST P-512 curve.
4. **Request SEV-SNP attestation report.** The verifier will request an attestation report from the prover. The request contains a nonce and the verifiers public key. The prover requests the attestation report, including the nonce and the provers public key.
5. **Verify attestation report.** The attestation response is verified by the verifier, by validating the signature of the attestation against the VCEK public key. And ensures the correctness of the attestation by checking the nonce and the included Kata policy checksum. Also, it extracts the provers public key from the attestation.
6. **Compute shared key.** Both verifier and prover compute the shared key using their private key and the other party's public key. Which will ensure both parties have the same shared key without the key itself being exchanged over the network.

Outcome: the verifier successfully validated the running workload at the verifier, and a shared key is created that is only known to the applications communicating with each other.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

6.1.3 Usage phase

This section describes the last of the three stages of remote attestation., the usage phase, which includes data exchange and result exchange, described below.

6.1.3.1 Data exchange

This scenario requires that the remote attestation scenario (see 6.1.2) has been completed, and, therefore, has as prerequisite that a shared key has been successfully exchanged between the verifier and prover. The aim of the data exchange scenario is to exchange data (encrypted with the shared key), thereby sending data from the verifier to the prover, so that the latter can process the data. The following steps are included:

7. **Request data.** The prover creates a request for data, optionally, encrypting the request with the shared key in case the request itself contains sensitive information.
8. **Process data request.** The verifier parses the request, optionally decrypting the request with the shared key. And subsequently, the right information is gathered to fulfill the request.
9. **Encrypt data response.** The verifier encrypts the data gathered for the request in a data response using the shared key.
10. **Decrypt data response.** The prover decrypts the data response with the shared key.
11. **Data usage.** The prover uses the data via their business logic.

Outcome: the prover received data from the verifier inside the trusted workload and is able to act on this data within the trusted workload pod.

6.1.3.2 Result exchange

This scenario requires the execution of the remote attestation (see 6.1.2) *s*cenario, and, therefore, has as prerequisite that a shared key has been successfully exchanged between the verifier and prover. The aim is to exchange the results of the trusted workload from the prover to the verifier. This includes the following steps:

1. **Request data processing.** The verifier creates a request for data processing, optionally encrypting the request with the shared key in case the request itself contains sensitive information.
2. **Process request for data processing.** The prover parses the request, optionally decrypting the request with the shared key. And subsequently executes the data processing within the trusted workload.
3. **Encrypt result response.** The prover encrypts the results of the data processing in a data processing response using the shared key.
4. **Decrypt result response.** The verifier decrypts the data processing response using the shared key.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

5. **Result usage.** The verifier uses the data processing response via their business logic.

Outcome: the verifier received the result of data processing that was executed at the prover, and is able to act on this data.

6.2 Logical view

Since remote attestation is not a software component but primarily describes the protocol both the verifier and prover should comply to, the logical view is described differently. In Figure 40, an overview of the logical steps is shown associated to the scenarios described above.

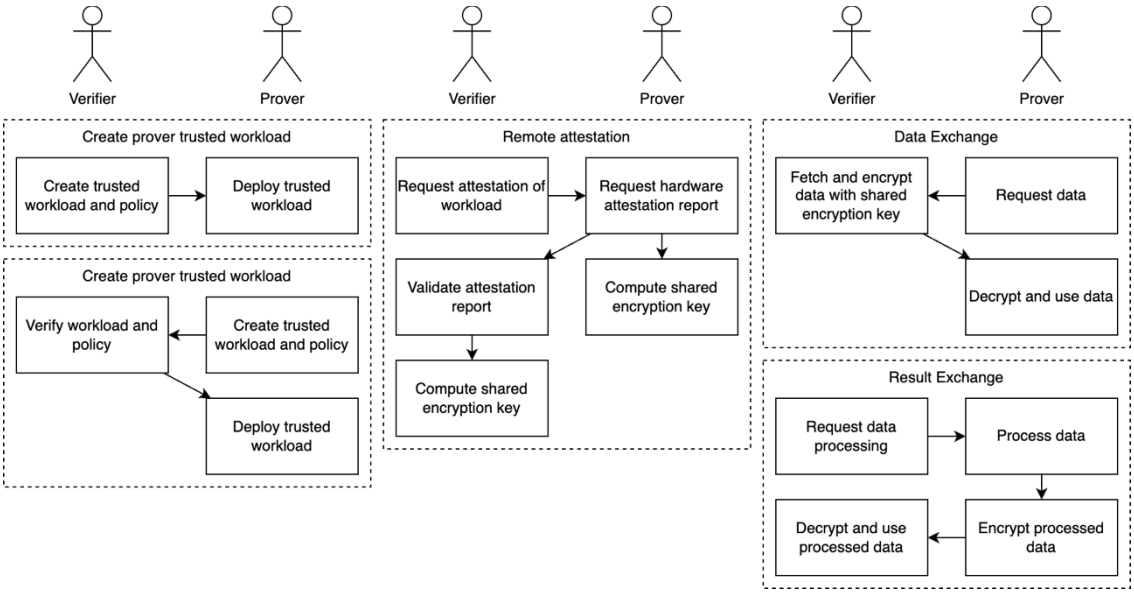


Figure 40: Logical view for each of the phases: preparation phase (left), remote attestation phase (middle), usage phase (right)

The steps related to the preparation are likely to be manual steps that the verifier and prover execute in their development processes. The result of these steps will ensure that both parties understand the trusted workload and its associated policy. Which can be exchanged during either the negotiation phase or the transfer phase in the Dataspace Protocol. For usage in negotiation, the expectation is that the workload is stable and won't change too often, since it is embedded in a negotiated contract between the two parties and changes in the workload or policy will require a new negotiation. Usage in the transfer phase on the other hand, will likely require an administrator to manually approve the transfer as the policy will be included in the parameters of the transfer.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

The Diffie-Hellman key exchange executed during the remote attestation phase allows for more flexible communication patterns to be executed between the two parties. As the verifier must have assurances that the communication after the remote attestation is still with the same application to prevent the prover to direct the communication to an untrusted application. This relies on the assumption that the workload and its policy doesn't allow for the shared key to be extracted from the trusted workload. Without the use of a shared key, the communication between the verifier and prover must be executed in the same channel as the remote attestation which would hinder the usage in compute clusters with ingress controllers or reverse proxies in front of the cluster that handle the traffic and TLS termination.

6.3 Process view

6.3.1 Preparation

The preparation of a trusted workload primarily involves the entity that develops the application, either the verifier or the prover. In case the verifier wants to securely deploy a trusted workload in the security domain of the prover, for instance to deploy an IP-sensitive algorithm that acts on data from the prover. Or when the prover wants to create a trusted workload that the verifier has enough trust in to share data from the verifier to the prover, for instance to perform analysis with an algorithm of the prover on data from the verifier and the verifier wants to have assurances that the data won't leak out of the trusted workload.

In both scenarios, the process for creating the trusted workload is the same:

1. Create an application containing business logic.
2. Package the application in a Docker container, optionally signing the image using Docker Content Trust.
3. Create the Kubernetes Pod template and determine which elements should be configurable. Especially when the workload is created by the prover, for instance links to the data and with the control and/or data plane.
4. Create the Kata Containers agent policy

The Kata Containers agent policy is the primary element for ensuring trust in the created application. Containing all the rules related to the trusted workload. These rules contain restrictions on the container itself but also all the related interactions a user or application may have with the pod, a non-exhaustive list of rules that can be set:

- Creation of containers, including all relevant rules like what image can be used but also what other parameters can be changed like the command of the container or environment variables.
- Limit standard in and out, resulting in logs not viewable for administrators.
- Limit execution of processes in the containers.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

- Limit file copy from and to the containers.
- Limit the ability of updating, pausing, resuming, or deleting the workload.

For the deployment of a trusted workload, the policy must be included as an annotation in the pod configuration together with setting the right Kata containers runtime class. This ensures that the containers in the pod will be executed inside a UVM with the policy provided.

6.3.2 Remote attestation

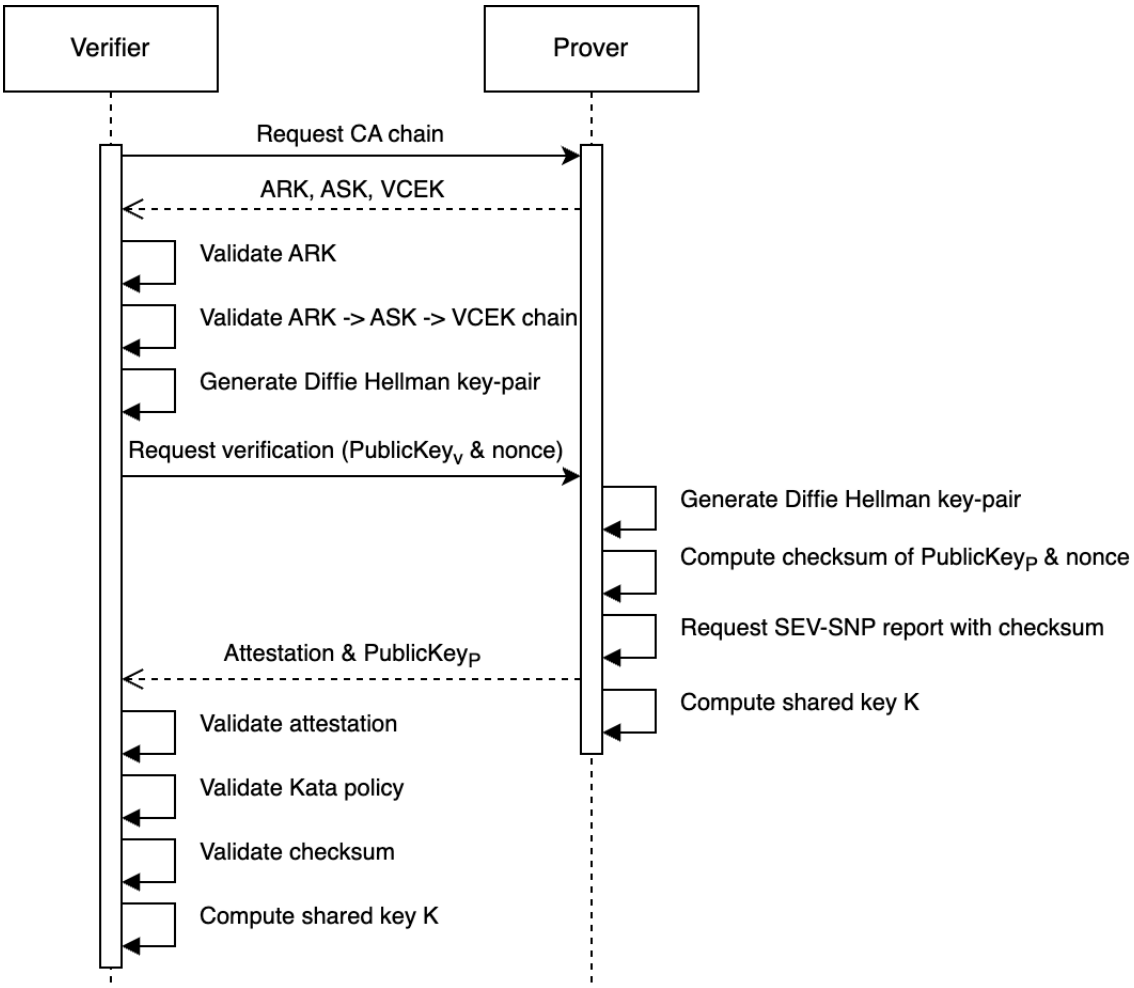


Figure 41: Remote attestation sequence diagram

1. Request CA chain of certificates used for attestation.
2. Retrieve the CA chain. For AMD SEV-SNP, these are: the AMD Root Key (ARK), the AMD Signing Key (ASK), and the Versioned Chip Endorsement Key (VCEK).

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

3. The verifier verifies the ARK certificate against the published certificate from AMD[20].
4. The verifier validates the chain of certificates, ensuring the ASK is signed by the ARK and the VCEK is signed by the ASK.
5. The verifier generates a private and public key for a Diffie-Hellman key exchange, based on the NIST P-512 curve.
6. The verifier requests a verification with a nonce and its public key.
7. The prover generates a private and public key for a Diffie-Hellman key exchange, based on the NIST P-512 curve.
8. The prover computes the SHA-512 checksum of the nonce concatenated with its public key.
9. The prover requests an SEV-SNP report, with as Report Data the computed checksum.
10. The attestation report together with the public key of the prover are returned.
11. The prover computes the shared key K based on its private key and the verifiers public key.
12. The verifier validates the signature in the attestation with the VCEK.
13. The verifier validates the Host Data in the attestation with the SHA-256 checksum of the Kata agent policy.
14. The verifier computes the SHA-512 checksum of the nonce concatenated with the public key of the prover. And verifies it against the Report Data field in the attestation.
15. The verifier computes the shared K based on its private key and the provers public key.

6.3.3 Usage

The usage process is split up into the two scenarios *Data Exchange* and *Result Exchange*.

[20] https://www.amd.com/en/developer/sev.html

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

6.3.3.1 Data Exchange

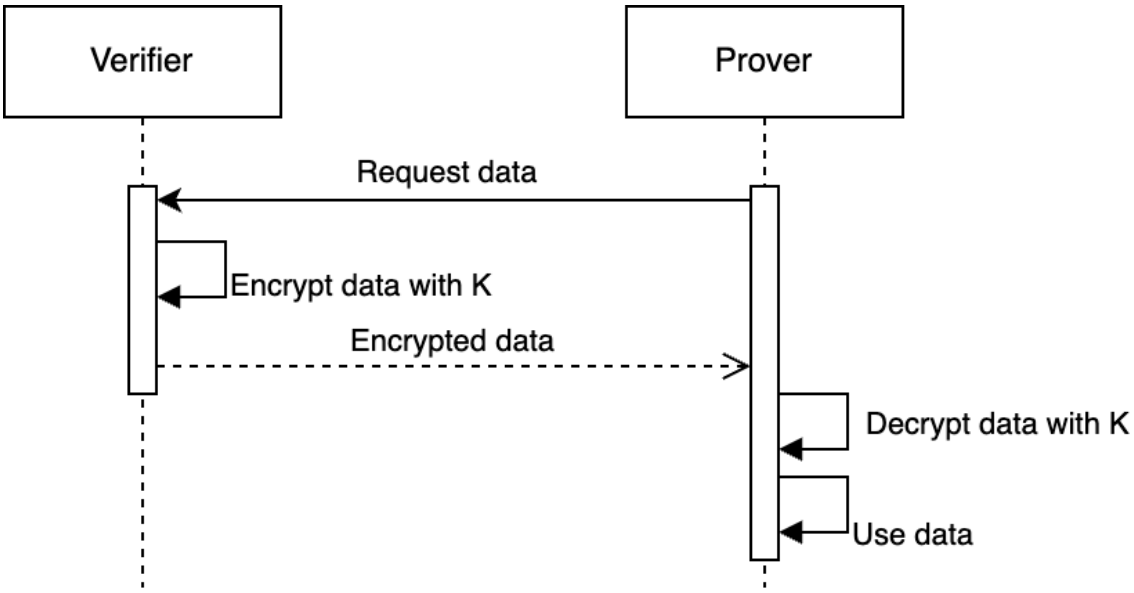


Figure 42: Data exchange sequence diagram

1. The prover requests the data from the verifier. Optionally, the request itself might be encrypted with the shared key K.
2. The verifier parses the request, if required decrypts the request. Fetches the relevant data and encrypts this data with the shared key K.
3. The verifier returns the encrypted data.
4. The prover decrypts the data.
5. The prover uses the data within its application.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

6.3.3.2 Result Exchange

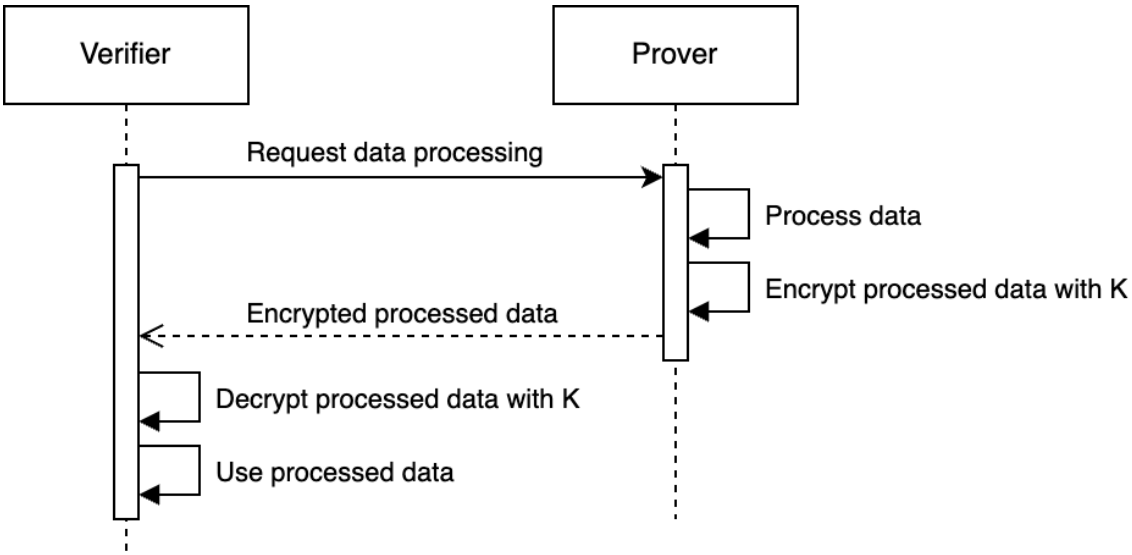


Figure 43: Result exchange sequence diagram

1. The verifier sends a request for data processing. Optionally, encrypting the request itself with the shared key K.
2. The prover parses the request, if required decrypts the request. And executes the data processing as requested.
3. The prover encrypts the processed data with the shared key K.
4. The prover returns the encrypted processed data.
5. The verifier decrypts the processed data.
6. The verifier uses the processed data within its application.

6.4 Development view

The development view is not applicable for remote attestation, since the protocols need to be embedded in the actual implementations using remote attestations.

6.5 Deployment view

Since the deployment of remote attestation solely relies on applications by the verifier and prover, no common components are deployed within the ENERSHARE data space.

A trusted third party, that checks trusted workloads and their policies to provide a list of trusted workloads, might be envisioned in a data space. This can be achieved via a simple connector that provides an interface for application developers to upload/reference their workload and policy and an interface for verifiers to request a list of trusted workloads.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

6.6 Conclusion

The remote attestation for full stack integrity evolved from plain virtual machine-based attestations in D4.2 based on the older IDS Communication Protocol v2 (IDSCPv2) towards a more scalable usage of full stack integrity. A major contribution to this change is the availability of the dataspace protocol, which allows the attestation of just the application handling the data due to the strict separation of control and data planes, contrary to the old way where all data flowed through the core container. This separation enabled the creation of an architecture for remote attestation that just focuses on the interactions between the data planes, leaving the control plane completely out of scope while keeping the security guarantees in place.

In addition, leveraging Confidential Containers in combination with Kata Containers allows the creation of trusted workload with reasonable security. The only confidential computing technology that would have increased security, by decreasing the size of the TCB, is enclave-based confidential computing, for instance Intel SGX. The reason not to choose enclave-based confidential computing is due to the added complexity of having the need to compile specific enclaves for each application. Especially in scenarios where the party deploying the trusted workload is not the same as the one that created the workload, the usability of specific enclaves would hinder the usage. Using VM-based confidential computing, with a specific and very small utility VM allows for any application, if it is containerized, to run without changes. The only decreasing factor for trust is the fact that the UVM must be trusted by both parties, since this UVM is often provided by the cloud provider there will be an external dependency. This could be mitigated by deploying a cluster manually with an UVM that both prover and verifier trust, but this would hinder the usability.

The scenarios that can be supported by confidential computing are much larger than the two usage scenarios, *data exchange* and *result exchange*, since as soon as a shared key has been exchanged this key is the anchor to the trusted exchange. An extension that could be envisioned to the *remote attestation* process is mutual attestation of both verifier and prover, resulting in both parties acting as both roles in the same flow. This can be achieved by executing the process twice, resulting in two shared keys. But this could also be optimized by creating just one shared key, requiring the verifier to provide its public key for the Diffie-Hellman exchange via an attestation report similarly to the way the prover provides its public key to the verifier.

All in all, the resulting architecture provides a solid basis for remote attestation of trusted workloads in a dataspace setting. While keeping the specific scenarios for dataspaces in mind and imposing only a limited set of requirements on both the prover and verifier.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

7 Conclusions

7.1 Framework for trust and sovereignty

The current ENERSHARE deliverable D4.3 described the final version of the framework for trust and sovereignty in the ENERSHARE project. This framework consisted of four building blocks, namely decentralized identity management (chapter 3), usage control enforcement (chapter 4), usage policy negotiation on the blockchain (chapter 5), and remote attestation for full stack integrity (chapter 6). This document outlined the successful integration of decentralized identity management between the ENERSHARE- and OMEGA-X project, the enforcement of usage control policies via data space connectors, the utilization of blockchain for usage policy notarization, and the project's advancements in remote attestation for full stack integrity.

Decentralized identity management was successfully implemented in the ENERSHARE- and OMEGA-X project, accepting each other's verifiable credentials (VCs) while maintaining their own Certificate Authorities (CAs). This ensured that each project was able to maintain its own unique identity- and security protocols while still achieving seamless interoperability. The upcoming proof of concept (PoC) in November will be a critical milestone in demonstrating the feasibility of cross-project credential acceptance.

Usage control enforcement focused on usage control and policy enforcement in the Eclipse Dataspace Connector (EDC). An EDC Extension for facilitating policy enforcement was developed and integrated with the EDC connector that allowed automatic negotiation of contracts using a front end. Earlier versions of trust and sovereignty components already included policy enforcement in TRUE and TSG connector implementations.

The development of usage control enforcement (in ENERSHARE) focused on an extension for the Eclipse Dataspace Connector (EDC) that facilitated policy enforcement. This extension was successfully integrated with the EDC and allowed for automatic contract negotiation via a user web-interface. Future work will involve testing the usage control module with real-case scenarios and further integration into the data space protocol, enhancing interoperability with the new version of TSG.

Usage policy negotiation on the blockchain focused on the development of the Blockchain Notarization Module. Initially intended to support Task 4.3 outcomes regarding usage control, the Blockchain Notarization Module evolved into a versatile tool during the ENERSHARE project that was applicable across different levels of the ENERSHARE stack. This module has been deployed in various pilots, supporting anonymization, certification of data sets, and enabling automated, decentralized asset trading. It was also integrated into the ENERSHARE Marketplace for notarizing auctions, demonstrating its broad applicability and effectiveness.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Remote attestation for full stack integrity has evolved from plain virtual machine-based attestations (as described in ENERSHARE deliverable D4.2) that were based on the older IDS Communication Protocol v2 (IDSCPv2) towards a more scalable usage of full stack integrity. A major contribution to this change was the availability of the dataspace protocol version 2024-1, which allowed the attestation of just the application handling the data due to the strict separation of control and data planes, contrary to the old way where all data flowed through the core container of the dataspace connector. This separation enabled the creation of an architecture for remote attestation that just focuses on the interactions between the data planes, leaving the control plane completely out of scope while keeping the security guarantees in place.

The resulting architecture provided a robust foundation for remote attestation in dataspaces, imposing minimal requirements on both provers and verifiers. Future enhancements may include mutual attestation, optimizing the process for greater security and efficiency.

7.2 Contribution to interoperability

Work package 4 provided several contributions to (technical) interoperability for energy dataspaces.

7.2.1 Dataspace Protocol

The ENERSHARE project acknowledged the Dataspace Protocol (see section 2.3) as important building block for interoperability between different dataspace connector implementations (including TSG, TRUE and EDC). This allows for a) reduced vendor lock-in and dependency on a single software implementation and b) improved interoperability with other dataspace, including the Int:net sister projects.

The adoption of the Dataspace Protocol (and thereby the transition from the previously used IDS protocol) is executed in a separate environment within the ENERSHARE project, parallel to the existing IDS Communication Guide based environment. This is to allow ENERSHARE partners to migrate to the Dataspace Protocol according to their own timeline. This is especially relevant for ENERSHARE partners that create services that are tightly integrated with the dataspace, e.g., the marketplace, the app store, and for which the transition to the Dataspace protocol has the highest impact.

7.2.2 Int:net System Use Case 1 (SUC1)

As part of task 4.2 and related to the results described in chapter 3 on decentralized identity management, the ENERSHARE project contributed to system use case (SUC) 1 of the Int:net project covering the onboarding process which includes identity management. On top of technical interoperability realized by the Dataspace Protocol, we've identified a solution for

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

decentralized identity management across dataspaces where each dataspace/project can maintain its identity and security protocols while still achieving interoperability.

7.2.3 IDSA Task Force Energy Interoperability

Furthermore we contributed to the activities of the IDSA Task Force on Energy Interoperability. On of the main outcomes of this task force is the position paper on interoperability in energy dataspaces[21] describing the challenges, state of the art, solutions and gap analysis. The ENERSHARE reference architecture, approach for trust and sovereignty (WP4) and input from other work packages are included in this position paper.

7.3 List of software components for final version

Work package 4 delivered several software components to be used and integrated in the ENERSHARE dataspace.

7.3.1 Connector implementations

The dataspace connector is the basis for trust and sovereignty. It provides secure peer-to-peer data exchange with IAA (Identification, Authentication, Authorization).

TNO Security Gateway (TSG)	
Description	IDS-based HTTP Multipart communication. See: Message Flows, Deployment. And the Playground to play around with the connector
Software details	Written in Kotlin, built into Docker images. Default deployment based on Kubernetes & Helm.
Codebase	• https://gitlab.com/tno-tsg/core-container • https://gitlab.com/tno-tsg/helm-charts/connector
ADDED Deployed for ENERSHARE dataspace	At the moment of writing this report, a total of 15 connectors are deployed in the ENERSHARE dataspace. To get an overview of the deployed connectors, see: https://daps.Enershare.dataspac.es/#connectors

TRUE Connector	
Description	IDS-based connector implementation consisting of an execution core container, a data app and usage control component.
Software details	Written in Java, built into Docker images. Docker compose and Kubernetes manifests available for deployment.

[21] https://internationaldataspaces.org/wp-content/uploads/dlm_uploads/IDSA-Position-paper-Energy-interoperability-framework-v0.9.pdf

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Codebase	• https://github.com/Engineering-Research-and-Development/true-connector
ADDED Extended functionality	Support for additional policy types, as described in Chapter 5

7.3.2 Identity provider (CA + DAPS)

The certificate authority (CA) issues identity certificates for connector instances by signing Certificate Signing Requests (CSRs) that have been handed in by valid connector instances. It revokes certificates that become invalid and, for higher trust levels, assure that private keys are properly stored in hardware modules (such as a TPM or HSM).

The DAPS component provides dynamic, up-to-date attribute information about Participants and Connectors in form of signed claims and embeds them into Dynamic Attribute Tokens (DATs).

TSG DAPS (includes CA)	
Description	Implementation of an IDS Dynamic Attribute Provisioning Service (DAPS) v2, combined with certificate authority capabilities to sign certificate signing requests (CSR)
Software details	Written in Typescript. Both backend and generic frontend application.
Codebase	• https://gitlab.com/tno-tsg/daps • https://tno-tsg.gitlab.io/docs/daps/
ADDED Deployed for ENERSHARE dataspace	Accessible via: https://daps.Enershare.dataspac.es/ See D8.2 – ENERSHARE Data Space (1st Technology Release) for more information.

7.3.3 Notarization service

Proof of Existence and API Layer	
Description	Smart contract running into an integrated Ethereum-based blockchain, and a Dapp providing a REST API Layer for easy notarization and verification of the integrity of data and documents.
Software details	Written in Solidity and Javascript, built into Docker images. Docker compose available for deployment.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Codebase	https://gitlab.com/dt-iot//proof-of-existence
Deployed for ENERSHARE dataspace	Designed for the Usage Control to enforce policy integrity. Shared also with the Slovenian pilot. Indirectly affecting the Energy dataspace, as a supporting tool.

7.4 Future research

This chapter described the conclusions of the final version of the framework for trust and sovereignty in the ENERSHARE project. We believe that future work will involve several key areas:

1. **Proof of Concept for Decentralized Identity Management**: The PoC scheduled for November will be crucial in validating cross-project credential acceptance, paving the way for broader adoption and implementation.
2. **Real-Case Scenario Testing for Usage Control**: The usage control module will be tested with real-case scenarios to ensure its effectiveness and reliability. Further integration with the data space protocol and the new TSG version will be required.
3. **Expansion of Blockchain Notarization**: The Blockchain Notarization Module will continue to be refined and expanded, ensuring its applicability across more pilots and use cases, particularly in the ENERSHARE Marketplace.
4. **Optimization of Remote Attestation**: Future extensions to the remote attestation process will focus on achieving mutual attestation, enhancing security by allowing both verifiers and provers to act in dual roles within a single flow.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831